Q Corresponding Ethernet Interface Module

MITSUBISHI

User's Manual

(Basic)

Mitsubishi Programmable Logic Controller

MELSEC-Q

QJ71E71-100
QJ71E71-B5
QJ71E71-B2

• SAFETY PRECAUTIONS •

(Always read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.
The instructions given in this manual are concerned with this product. For the safety instructions of the programmable controller system, please read the user's manual of the CPU module to use.
In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".

⚠ DANGER	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Note that the ⚠CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

[Design Precautions]

⚠ DANGER

- For details on the operating status of each station when a communication abnormality occurs in the data link, see the manual for each data link. Erroneous outputs and malfunctions may lead to accidents.
  Not doing so can cause an accident due to false output or malfunction.
- To prevent malfunctions of the PLC system that may be caused by illegal e-mails from the outside, take a proper countermeasure (such as virus detection) so that illegal e-mails are not received by the mail server of this module.
- If it is necessary to ensure the security of the PLC system against unauthorized access from external devices via the Internet, appropriate measures must be incorporated by the user.
- Then controlling a running PLC (modifying data) by connecting peripheral devices to the CPU module or connecting a personal computer to the intelligent function module, configure an interlocking circuit in a sequence program so that the safety of the overall system is always maintained. Also, before performing other control operations (program modifications and operation status modifications (status control)) on the running PLC, be sure to read the manual carefully and thoroughly confirm the safety.
  Especially in the above mentioned control operations that are performed from an external device to a remote PLC, any problems on the PLC side may not be dealt with promptly due to abnormal data communication. In addition to configuring an interlocking circuit in a sequence program, determine how the system handles data communication abnormalities, etc. between the opposite devices and the PLC CPU.

[Design Precautions]

⚠ DANGER

- Do not write any data in the "system area" of the buffer memory of the intelligent function module. Also, do not output (turn on) the "use prohibited" signal, which is one of the output signals from the PLC CPU to the intelligent function module. If data is written to the "system area" or the "use prohibited" signal is output, there is a risk that the PLC system may malfunction.

⚠ CAUTION

- Do not bundle the control wires and the communication cables with the main circuit and the power wires, and do not install them close to each other. They should be installed at least 100 mm (3.94 in.) away from each other. Failure to do so may generate noise that may cause malfunctions.
- When the status control (remote RUN/STOP, etc) of the PLC CPU is performed from the external device, select the "Always wait for OPEN" parameter set by an user in advance. (Select with the initial timing setting in the operational setting.) If "Do not wait for OPEN" is selected the communication line at remote STOP is closed. The communication line cannot be reopened on the PLC CPU side after that, and the remote RUN from the external device cannot start either.

[Installation Precautions]

⚠ DANGER

- Use the PLC in the operating environment that meets the general specifications described in the user's manual of the CPU Module to use. Using the PLC in any other operating environments may cause electric shocks, fires or malfunctions, or may damage or degrade the module.
- While pressing the installation lever located at the bottom of module, insert the module fixing tab into the fixing hole in the base unit until it stops. Then, securely mount the module with the fixing hole as a supporting point.
  If the module is not installed properly, it may cause the module to malfunction, fail or fall off. Secure the module with screws especially when it is used in an environment where constant vibrations may occur.
- Be sure to tighten the screws using the specified torque. If the screws are loose, it may cause the module to short-circuit, malfunction or fall off. If the screws are tightened excessively, it may damage the screws and cause the module to short-circuit, malfunction or fall off.
- Before mounting or dismounting the module, make sure to shut off all phases of the external power supply. Failure to do so may damage the module.
- Do not directly touch the conducting parts and electronic parts of the module.
  This may cause the module to malfunction or fail.

[Wiring Instructions]

⚠ CAUTION

- Use crimp-contact, pressure-displacement or soldering to wire the connectors for external connections properly using the manufacturer-specified tools.
  If the connection is incomplete, it may cause the module to short circuit, catch fire, or malfunction.
- Do no connect the AUI cable when the power to the station in which the module is loaded is on.
- Make sure to place the communication and power cables to be connected to the module in a duct or fasten them using a clamp. If the cables are not placed in a duct or fastened with a clamp, their positions may be unstable or moved, and they may be pulled inadvertently.
  This may damage the module and the cables or cause the module to malfunction because of faulty cable connections.
- Tighten the terminal screws using the specified torque. If the terminal screws are loose, it may cause the module to short-circuit, malfunction or fall off. If the terminal screws are tightened excessively, it may damage the screws and cause the module to short-circuit, malfunction or fall off.
- When disconnecting the communication and power cables from the module, do not pull the cables by hand. When disconnecting a cable with a connector, hold the connector to the module by hand and pull it out to remove the cable. When disconnecting a cable connected to a terminal block, loosen the screws on the terminal block first before removing the cable. If a cable is pulled while being connected to the module, it may cause the module to malfunction or damage the module and the cable.
- Be careful not to let any foreign matter such as wire chips get inside the module. They may cause fire, as well as breakdowns and malfunctions of the module.
- A protective sheet is pasted on the upper part of the module in order to prevent foreign matter such as wire chips to get inside the module while wiring.
  Do not remove this protective sheet during wiring work. However, be sure to remove the protective sheet before operating the module to allow heat radiation during operation.
- Make sure to install the attached terminal covers on the product before turning the power back on and starting the operation after wiring work. Failing to install the terminal covers may result in malfunctions.
- Solder the coaxial cable connectors properly. Incomplete soldering may result in malfunctions.

A - 3

[Setup and Maintenance Precautions]

⚠ CAUTION
• Never disassemble or modify the module. This may cause breakdowns, malfunctions, injuries or fire. • Before mounting or dismounting the module, make sure to shut off all phases of the external power supply. Failure to do so may cause the module to breakdown or malfunction. • Do not mount/remove the module onto/from base unit more than 50 times (IEC61131-2-compliant), after the first use of the product. Failure to do so may cause the module to malfunction due to poor contact of connector. • Do not touch the terminals while the power is on. Doing so may cause electric shocks or malfunctions. • Before cleaning the module or retightening the terminal screws and the module mounting screws, make sure to shut off all phases of the external power supply. Failure to do so may cause the module to breakdown or malfunction. If the screws are loose, it may cause the module to short-circuit, malfunction or fall off. If the screws are tightened excessively, it may damage the screws and cause the module to short circuit, malfunction or fall off. • Always make sure to touch the grounded metal to discharge the electricity charged in the body, etc., before touching the module. Failure to do so may cause a failure or malfunctions of the module.

[Operating Precautions]

⚠ CAUTION
• Please read the user's manual carefully and confirm the safety thoroughly before proceeding to perform control operations of the PLC (especially, modifications of data, programs and operation status (status control)) that is running by connecting personal computers, etc. to the intelligent function module. Incorrect modifications of data, programs and operating status may cause system malfunctions, damages to the machines, or accidents.

[Precautions When Disposing of This Product]

⚠ CAUTION
• Dispose of this product as an industrial waste.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Dec., 1999	SH (NA) -080009-A	First Edition
Oct., 2000	SH (NA) -080009-B	Reflect the contents of the function version B. Put Windows® base software products together from Mitsubishi Programmable Logic Controller MELSEC series to Mitsubishi integrated FA software MELSOFT series. Standardize the name from software package (GPP function) to product name (GX Developer). Correction Entire manual (change MELSECNET/10H to MELSECNET/H), SAFETY PRECAUTIONS, Contents, About Manuals, The Manual's Usage and Structure (Structure of this manual (2)), About the Generic Terms and Abbreviations, Section 1.1, 1.2, 1.3, Section 2.1, 2.2, 2.3, 2.5, 2.6, Section 3.1, 3.2, 3.5, 3.6, 3.7, 3.8 (2), Section 4.1.1, 4.2, 4.3, 4.4.1, 4.5 (entire), 4.5.1 (Table), 4.6, 4.7, 4.8, 4.9 (entire), Section 5.2.1, 5.2.2, 5.3, 5.5, 5.6 (entire), 5.7.2, 5.8, Chapter 6 (entire), Section 7.1, 7.2, 7.3.1, 7.3.2, 7.4.2, 7.5.2, Chapter 8, Section 8.1, 8.2, 8.3.1, 8.3.2, 8.5.1, 8.6.2, Section 9.2.3, Section 10.2 to 10.8, Chapter 11, Section 11.1.1, 11.1.2 (2), 11.2, 11.2.2, 11.2.4, 11.3 (5) (6), 11.3.1 to 11.3.3, 11.4, 11.4.4, Appendix 1 (entire), Appendix 2 (entire), Appendix 3, Appendix 8 (entire), Appendix 11 Addition Entire manual (add the explanation on MELSECNET/H remote I/O station), The Manual's Usage and Structure (2) (e), Section 5.4, Section 11.2, 11.2.1, 11.2.3, 11.3.1 (error code 63H), 11.3.3 (error code 0063H, C086H, C087H, C0DAH, C0DBH, C119H, C200H to C205H)
Jun., 2001	SH (NA)-080009-C	Added the description of the model QJ71E71-100 Ethernet interface module. Additional model QJ71E71-100 Correction Conformation to the EMC Directive and the Low Voltage Instruction, The Manual's Usage and Structure, About the Generic Terms and Abbreviations, Product Configuration, Section 1.2, 1.3, 1.4 (Figure), Section 2.1, 2.2, 2.3 (POINT), 2.4, 2.5, 2.7, Chapter 3 (entire), Section 4.2, 4.3, 4.4, 4.7, 4.8.1, Section 5.1 (Figure), 5.3 (1) (Figure), 5.4, 5.5, 5.6 (POINT), 5.6.1, 5.6.2, 5.8 (2) (3), 5.9.1 (1) (2), 5.9.3 (Figure), 5.9.5, 5.9.6 (3), 5.9.7, Section 6.2, Section 7.1, 7.2 (Figure), 7.3.1 (Figure), 7.3.2 (2) (Figure), 7.5.2, Section 8.2 (Figure), 8.3.1 (Figure), 8.3.2 (2) (Figure), 8.5.1, 8.6.2, Section 9.2.3 (2) (Figure), Section 10.1 (POINT), 10.6, 10.8, Section 11.1.1, 11.1.2, 11.2.3, 11.3 (7) (Figure), 11.3.3 (error codes C113H, C114H, and C14EH), 11.4 (POINT), 11.4.6 (Figure), Appendix 1.1, Appendix 2.1, 2.2 (2), Appendix 8.3, Appendix 11

A - 5

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Jun., 2001	SH (NA)-080009-C	Addition Section 1.2 (5) (6), Section 2.2 (1), 2.6, Section 5.4.2, Section 11.3 (6), 11.3.2 (error code 1FH), 11.3.3 (error codes C0F7H and C300H), Appendix 4, Appendix 7, Appendix 9, Appendix 11
Oct., 2001	SH (NA)-080009-D	Correction Section 1.2 (4) (7), 1.3, 1.4 (1), Section 2.2 (POINT), 2.5 (1), 2.7 (1), Section 3.1, 3.5, 3.8, Section 4.1.1 (2), 4.3, 4.4.1 (POINT), 4.7 (5), Section 5.2.2 (REMARK), 5.2.3, 5.4.2, 5.5 (Table), 5.6.1, 5.6.2, 5.6.3, Section 10.1, Section 11.2.2 (2), 11.2.4, 11.3.3, Appendix 1.1, Appendix 8.2, Appendix 9 Addition Section 10.9
Mar., 2002	SH (NA) -080009-E	Correction About the Generic Terms and Abbreviations, Section 1.2 (7), Section 2.1 (1), 2.7 (1), Section 3.7 (Table), Section 5.1 (Figure), 5.2.2 (REMARK), 5.2.3, Section 7.5.1, Section 10.9, Section 11.3.1 (Table), 11.3.3 (REMARK), 11.4.2 (Figure), Appendix 1.1 (2) (Table), Appendix 2.2 (2) (a), Appendix 8 Addition Section 11.3.3 (error code C1BAH), 11.4 *2, Appendix 4 (4)
Apr., 2003	SH (NA)-080009-F	Additional model QJ71E71-B5 Deleted model QJ71E71 Correction SAFETY PRECAUTIONS, About the Generic Terms and Abbreviations, Section 1.2 (1) (5) (6), 1.3, 1.4 (1), Section 2.1, 2.2, 2.5, 2.6, 2.7, Section 3.1, 3.4, 3.8, Section 4.1, 4.3, 4.4, 4.5, 4.6, Section 5.2.2 (REMARK) (5), 5.2.3 (REMARK), 5.3, 5.4.2 (1)(c), 5.4.3, 5.5 (6), 5.6 (2) (POINT), 5.9.1, 5.9.6 (3) (6), Section 6.2, Section 10.8, 10.9, Section 11.1.1, 11.2.2 (2), 11.3, 11.4.5 (Figure), Appendix 1.1, Appendix 2, Appendix 4, Appendix 7, Appendix 10, Appendix 11 Addition Section 11.2.1 (POINT), 11.3.3 (error code C0B2H, C0E0H to C0EFH, C171H to C17FH), Appendix 9
Jul., 2003	SH (NA)-080009-G	Correction The Manual's Usage and Structure, Section 1.4 (1), Section 2.2 (1), 2.7, Section 3.8, Section 4.7, Section 5.2.2 (REMARK), 5.5 (POINT), 5.6.1, 5.6.2, Section 7.5.2, Section 8.6.2, Section 11.4, Appendix 2, Appendix 8.1, Appendix 11, Appendix 12

A - 6

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Sep., 2003	SH (NA)-080009-H	Correction Section 1.4 (1), Section 3.8, Section 5.2.3, Appendix 2.1, Appendix 8
Jun., 2004	SH (NA)-080009-I	Addition of the description of function version D Correction About the Generic Terms and Abbreviations, Section 1.1 (7), Section 1.3, Chapter 2 (entire), Section 3.6, Section 3.8, Section 4.5, Section 4.6, Section 5.3, Section 5.5, Section 5.6 (2) (POINT), Section 5.8 (POINT), Section 6.1.4, Section 10.9, Appendix 1, Appendix 4, Appendix 11, Appendix 12 Addition Section 1.1 (9), Section 5.10, Section 5.11, Section 11.3.3, Section 11.4.7, Appendix 8.2

Japanese Manual Version SH-080004-N

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 1999 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Thank you for purchasing the MELSEC-Q series PLC.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the Q series PLC you have purchased, so as to ensure correct use.
Please forward a copy of this manual to the end user.

CONTENTS (This manual)

SAFETY PRECAUTIONS ..... A- 1
REVISIONS ..... A- 5
CONTENTS ..... A- 8
About Manuals A-16
Conformation to the EMC Directive and Low Voltage Instruction A-16
The Manual's Usage and Structure A-17
About the Generic Terms and Abbreviations A-21
Product Configuration A-22

1 OVERVIEW 1- 1 to 1-15

1.1 Overview of the Ethernet Module 1- 1
1.2 Features of the Ethernet Module 1- 2
1.3 Additional Functions in Function Version B or Later 1-12
1.4 Software Configuration 1-14

2 SYSTEM CONFIGURATIONS 2- 1 to 2-19

2.1 Applicable Systems 2- 1
2.2 Devices Required for Network Configuration 2- 3
2.3 For Use in Multiple CPU System 2- 8
2.4 For Use with Q00J/Q00/Q01CPU 2-10
2.5 For Use with Q12PRH/Q25PRHCPU 2-11
2.6 For Use at MELSECNET/H Remote I/O Station 2-14
2.7 Checking the Function Version and Serial No. 2-18

3 SPECIFICATIONS 3- 1 to 3-27

3.1 Performance Specifications 3- 1
3.2 Data Codes for Communication 3- 3
3.3 Relationship between the External Devices and Additional Functions for Each Communication Function 3- 5
3.4 Ethernet Module Function List 3- 6
3.5 Dedicated Instruction List 3- 8
3.6 List of GX Developer Setting Items for Ethernet Modules 3- 9
3.7 List of Input/Output Signals to/from the PLC CPU 3-10
3.8 List of Applications and Assignments of the Buffer Memory 3-12

4 SETTINGS AND PROCEDURES PRIOR TO OPERATION 4- 1 to 4-25

4.1 Loading and Installation 4- 1
4.1.1 Handling precautions 4- 1
4.1.2 Installation environment 4- 2
4.2 Settings and Procedures Prior to Starting the Operation 4- 3
4.3 Components of the Ethernet Module 4- 5
4.4 Connecting to the Network 4- 7
4.4.1 Connecting to the 10BASE-T/100BASE-TX network 4- 8
4.4.2 Connecting to the 10BASE5 network 4- 9
4.4.3 Connecting to the 10BASE2 network 4-10
4.5 Settings from GX Developer 4-12
4.5.1 I/O assignment setting 4-12
4.5.2 Other settings 4-13
4.6 Network Parameters Setting the Number of MNET/10H Ethernet Cards 4-16
4.7 Operational Settings 4-19
4.8 Self-Diagnostic Tests 4-22
4.8.1 Self refrain test 4-22
4.8.2 Hardware test (H/W Test) 4-23
4.9 Maintenance and Inspection 4-24
4.9.1 Maintenance and inspection 4-24
4.9.2 Mounting and dismounting the module 4-25

5 COMMUNICATION PROCEDURE 5- 1 to 5-111

5.1 Overview of the Communication Procedure 5- 1
5.2 Initial Processing 5- 3
5.2.1 Initial processing 5- 3
5.2.2 Initial settings 5- 4
5.2.3 Re-initial Processing 5-10
5.3 Router Relay Parameter 5-16
5.4 Confirming the Completion of the Initial Processing 5-21
5.4.1 PING test using GX Developer (Via Ethernet board) 5-21
5.4.2 PING test using GX Developer (Via CPU) 5-26
5.4.3 Loop back test using GX Developer 5-29
5.4.4 PING command (Personal computer → Ethernet module) 5-34
5.4.5 Loop back test (Communication using the MC protocol) 5-35
5.5 Open Settings 5-36
5.6 Open Processing/Close Processing of the Connection 5-43
5.6.1 Active open processing/close processing 5-45
5.6.2 Passive open processing/close processing 5-52
5.6.3 UDP/IP open processing/close processing 5-60
5.7 Pairing Open 5-63
5.7.1 Pairing open 5-63
5.7.2 Example of pairing open settings from GX Developer 5-64
5.8 Automatic Open UDP Port 5-66

5.9 Corresponding with the QCPU Remote Password Function 5-68
5.9.1 Data communication when a remote password is set 5-69
5.9.2 Precautions when using the remote password check function 5-76
5.9.3 Data communication procedure 5-77
5.9.4 When the remote password unlock processing or lock processing is completed abnormally 5-79
5.9.5 How to set the target connection for the remote password check 5-81
5.9.6 Buffer memory for the remote password check function 5-83
5.9.7 Data communication when the remote password check is set 5-87
5.10 Hub Connection Status Monitor Function 5-88
5.11 Configuring a Network in Redundant System (Redundant System Support Function) 5-89
5.11.1 Issue of System Switching Request to Control System CPU 5-89
5.11.2 Communication path bypass function 5-95
5.11.3 Redundant settings 5-97
5.11.4 Buffer memory for redundant system support function 5-99
5.11.5 Data communication for using Ethernet module in redundant system 5-100

6 COMMUNICATION USING THE MC PROTOCOL 6- 1 to 6- 5

6.1 Data Communication Function 6- 1
6.1.1 Accessing the PLC CPUs using the MC protocol 6- 1
6.1.2 Message format and control procedure for data communication 6- 2
6.1.3 PLC CPU setting for performing data communication 6- 3
6.1.4 Compatibility with multiple CPU system or redundant system 6- 4
6.1.5 Support for the QCPU remote password function 6- 4
6.2 Utilizing the MX Component 6- 5

7 FIXED BUFFER COMMUNICATION (WITH THE PROCEDURE EXIST CONTROL METHOD) 7- 1 to 7-22

7.1 Control Method 7- 1
7.2 Sending Control Method 7- 3
7.3 Receiving Control Method 7- 5
7.3.1 Receive processing with the main program (dedicated instruction: BUFRCV) 7- 5
7.3.2 Receive processing with an interrupt program (dedicated instruction: BUFRCVS) 7- 7
7.4 Data Format 7-12
7.4.1 Header 7-12
7.4.2 Application data 7-13
7.5 Programming 7-17
7.5.1 Precautions when creating programs 7-17
7.5.2 Fixed buffer communication program example (with the procedure exist control method) 7-18

8 FIXED BUFFER COMMUNICATION (WITH THE NO PROCEDURE CONTROL METHOD) 8- 1 to 8-21

8.1 Control Method 8- 1
8.2 Sending Control Method 8- 4
8.3 Receiving Control Method 8- 6
8.3.1 Receive processing with the main program (dedicated instruction: BUFRCV) 8- 6
8.3.2 Receive processing with an interrupt program (dedicated instruction: BUFRCVS) 8- 8
8.4 Data Format 8-10

8.5 Simultaneous Broadcast Using UDP/IP 8-11
8.5.1 Sending by simultaneous broadcasting 8-11
8.5.2 Receiving by simultaneous broadcasting 8-13
8.5.3 Precautions when using the simultaneous broadcast function 8-16
8.6 Programming 8-17
8.6.1 Precautions when creating programs 8-17
8.6.2 Fixed buffer communication program example (with the no procedure control method) 8-18

9 COMMUNICATION USING THE RANDOM ACCESS BUFFER 9- 1 to 9-16

9.1 Control Method 9- 1
9.1.1 Control method by read requests from an external device 9- 3
9.1.2 Control method by write requests from an external device 9- 4
9.2 Data Format 9- 5
9.2.1 Header 9- 5
9.2.2 Application data 9- 6
9.2.3 Examples of the command/response formats 9-11
9.3 Physical and Logical Addresses of the Random Access Buffer 9-15
9.4 Precautions when Creating Programs 9-16

10 DEDICATED INSTRUCTIONS 10- 1 to 10-26

10.1 Dedicated Instruction List 10- 1
10.2 BUFRCV Instruction 10- 2
10.3 BUFRCVS Instruction 10- 5
10.4 BUFSND Instruction 10- 8
10.5 CLOSE Instruction 10-11
10.6 ERRCLR Instruction 10-14
10.7 ERRRD Instruction 10-17
10.8 OPEN Instruction 10-19
10.9 UINI Instruction 10-23

11 TROUBLESHOOTING 11- 1 to 11-63

11.1 How to Check Errors Using LED Displays 11- 2
11.1.1 Checking error display 11- 2
11.1.2 How to turn off COM.ERR LED and to read/clear error information 11- 4
11.2 How to Check an Error Through GX Developer 11- 5
11.2.1 Ethernet diagnostics 11- 6
11.2.2 System monitor 11- 8
11.2.3 Buffer memory that can be monitored with the GX Developer diagnostic function 11-10
11.2.4 Checking the error information by the buffer memory batch monitoring function 11-12
11.3 Error Code List 11-13
11.3.1 End codes (Complete codes) returned to an external device during data communication 11-22
11.3.2 Abnormal codes returned during communication using A compatible 1E frames 11-24
11.3.3 Error codes stored in the buffer memory 11-25

11.4 Troubleshooting Flowchart 11-42
11.4.1 Sending errors during fixed buffer communication (common to procedure exist and no procedure) 11-45
11.4.2 Receiving errors during fixed buffer communication (common to procedure exist and no procedure) 11-47
11.4.3 Errors during random access buffer communication 11-50
11.4.4 Errors in communication using the MC protocol 11-52
11.4.5 Sending errors during e-mail communication 11-54
11.4.6 Receiving errors during e-mail communication 11-56
11.4.7 Error in redundant system 11-58

APPENDIX App- 1 to App-65

Appendix 1 Function Upgrade for the Ethernet Module ..... App- 1
Appendix 1.1 A comparison of functions of the Ethernet module ..... App- 1
Appendix 1.2 Precautions when updating the module from function version A to function version B or later ..... App- 3
Appendix 2 The QnA/A Series Module ..... App- 4
Appendix 2.1 Functional comparisons between the Ethernet modules and QnA/A series modules ..... App- 4
Appendix 2.2 Using the programs designed for QnA/A series modules ..... App- 6
Appendix 3 Installing the Ethernet Module on Existing Systems ..... App- 9
Appendix 4 Processing Time ..... App- 9
Appendix 5 ASCII Code List App-17
Appendix 6 References App-17
Appendix 7 External Dimensions App-18
Appendix 8 Program Examples App-20
Appendix 8.1 Program example for communication using the MC protocol -1 ..... App-21
Appendix 8.2 Program example of communication using the MC Protocol -2 ..... App-30
Appendix 8.3 Program example of communication using the MC Protocol -3 ..... App-41
Appendix 9 Communication Support Tool (MX Component) App-44
Appendix 9.1 Overview of MX Component App-44
Appendix 9.2 Usage Procedure of MX Component App-47
Appendix 10 Differences between the Ethernet and the IEEE802.3 ..... App-51
Appendix 11 ICMP Protocol Supported by the Ethernet Module App-51
Appendix 12 Setting Value Recording Sheets App-52

INDEX Index- 1 to Index- 3

(Related Manual-1) ··· Q Corresponding Ethernet Interface Module User's Manual (Application)
SH-080010-G

1 OVERVIEW
1.1 Overview
1.2 Additional Functions in Function Version B or Later

2 USING THE E-MAIL FUNCTION
2.1 E-mail Function
2.2 Configuration and Environment of the Applicable System
2.3 Precautions for Using the E-mail Function
2.4 E-mail Specifications
2.5 Processing Procedure of the E-mail Function
2.6 E-mail Setting from GX Developer
2.7 Sending/Receiving E-mail (Attached Files) by the PLC CPU
2.8 Sending E-mail (Main Text) by the PLC CPU
2.9 Sending E-mails Using the PLC CPU Monitoring Function

3 WHEN COMMUNICATING WITH MELSECNET/H, MELSECNET/10 RELAY
3.1 MELSECNET/H, MELSECNET/10 Relay Communication
3.2 Remote Station PLC Accessible Range and Stations
3.3 Settings for Accessing Other Stations
3.4 Procedure for Accessing Other Stations
3.5 Precautions for Accessing Other Stations

4 WHEN THE QCPU ACCESSES THE OTHER STATION PLC USING THE DATA LINK INSTRUCTION
4.1 Other Station Access with the Data Link Instruction
4.2 Precautions for Accessing Other Stations
4.3 Using the Data Link Instructions
4.4 Data Link Instructions
4.5 Data Sending/Receiving
4.6 Reading/Writing Word Devices of Other Stations (READ/WRITE)
4.7 Reading/Writing Word Devices of Other Stations (ZNRD/ZNWR)
4.8 Reading/Writing Clock Data, Remote RUN/Remote STOP (REQ)
4.9 Error Codes for Data Link Instructions

5 WHEN USING FILE TRANSFER FUNCTIONS (FTP SERVER)
5.1 File Transfer Functions
5.2 File Transferable Range
5.3 FTP Parameter Settings for File Transfer from GX Developer
5.4 Procedure and Required Processing on the External Device Side (FTP Client)
5.5 Precautions when Using the File Transfer Functions
5.6 FTP Commands

6 DEDICATED INSTRUCTIONS
6.1 Dedicated Instruction List
6.2 MRECV Instruction
6.3 MSEND Instruction
6.4 READ Instruction
6.5 RECV Instruction (for the Main Program)
6.6 RECVS Instruction (for Interrupt Programs)
6.7 REQ Instruction (Remote RUN/STOP)
6.8 REQ Instruction (Clock Data Read/Write)
6.9 SEND Instruction
6.10 SREAD Instruction
6.11 SWRITE Instruction
6.12 WRITE Instruction
6.13 ZNRD Instruction
6.14 ZNWR Instruction

A - 13

(Related Manual-2) ··· Q Corresponding Ethernet Interface Module User's Manual (Web function)
SH-080180-B

1 OVERVIEW
1.1 Overview

2 SYSTEM CONFIGURATIONS
2.1 System Configurations
2.2 Precautions for Using the Web Function

3 OPERATING PROCEDURE
3.1 General Procedure up to Communication Using the Web Function
3.2 How to Obtain and Set Up the Communication Library and the Sample Screen

4 VERIFYING THE OPERATION OF THE WEB FUNCTION USING A SAMPLE SCREEN
4.1 Web Function Items Available on the Sample Screen
4.2 Operating Procedure
4.3 Explanation of the Sample Screen
4.4 Example of Data Communication on the Sample Screen
4.5 Configurations of Files on the Sample Screen

5 EXAMPLE OF CREATING A FILE FOR ACCESSING THE PLC

6 COMMUNICATION LIBRARY FUNCTIONS

A - 14

(Related Manual-3) … Q Corresponding MELSEC Communication Protocol Reference Manual

SH-080008-F

1 OVERVIEW
- 1.1 Overview of the MELSEC Communication Protocol
- 1.2 Features of the MELSEC Communication Protocol

2 DATA COMMUNICATION USING THE MELSEC COMMUNICATION PROTOCOL
- 2.1 Types and Applications of Data Communication Frames
- 2.2 Accessible Range of Each Data Communication Frames
- 2.3 How to Read the Control Procedures of the MC Protocol
- 2.4 Access Timing of the PLC CPU Side
- 2.5 Setting Method for Writing to the PLC CPU during RUN
- 2.6 Accessing Other Stations
- 2.7 Precautions on Data Communication
- 2.8 Time Chart and Communication Time of the Transmission Sequence of the Serial Communication Module
- 2.9 Transmission Time When Accessing Other Stations Via MELSECNET/H, MELSECNET/10
- 2.10 Compatibility with Multiple CPU Systems
- 2.11 Compatibility with the Q00CPU, Q01CPU Serial Communication Function

3 WHEN COMMUNICATING USING THE QnA COMPATIBLE 3E/3C/4C FRAMES
- 3.1 Message Formats
- 3.2 List of Commands and Functions for the QnA Compatible 3E/3C/4C Frames
- 3.3 Device Memory Read/Write
- 3.4 Buffer Memory Read/Write
- 3.5 Reading from and Writing to the Buffer Memory of an Intelligent Function Module
- 3.6 PLC CPU Status Control
- 3.7 Drive Memory Defragmentation (for Other Station QnACPU)
- 3.8 File Control
- 3.9 Registering, Deleting and Reading User Frames: for Serial Communication Modules
- 3.10 Global Function: for Serial Communication Modules
- 3.11 Data Transmission to an External device (On-Demand Function): for Serial Communication Modules
- 3.12 Initializing the Transmission Sequence: for Serial Communication Modules
- 3.13 Mode Switching: for Serial Communication Module
- 3.14 Turning Off Displayed LEDs and Initializing Communication Error Information and Error Code: for Serial Communication Module
- 3.15 Turning Off the COM.ERR LED: for Ethernet Modules
- 3.16 Loopback Test
- 3.17 Registering or Canceling PLC CPU Monitoring: for Serial Communication Modules
- 3.18 Remote Password Unlock/Lock

4 WHEN COMMUNICATING USING THE QnA COMPATIBLE 2C FRAMES
- 4.1 Control Procedures and Message Formats
- 4.2 Contents of the Data Designation Items
- 4.3 List of Commands and Functions for QnA Compatible 2C Frames
- 4.4 Precautions on the Data Communication
- 4.5 Example of Data Communication Using QnA Compatible 2C Frames

5 WHEN COMMUNICATING USING THE A COMPATIBLE 1C FRAMES
- 5.1 Control Procedures and Message Formats
- 5.2 Device Memory Read/Write
- 5.3 Extension File Register Read and Write
- 5.4 Reading and Writing in the Buffer Memory of an Intelligent Function Module
- 5.5 Loopback Test

6 WHEN COMMUNICATING USING THE A COMPATIBLE 1E FRAMES
- 6.1 Message Formats and Control Procedures
- 6.2 List of Commands and Functions for A Compatible 1E Frames
- 6.3 Device Memory Read/Write
- 6.4 Extension File Register Read and Write
- 6.5 Reading and Writing in the Buffer Memory of an Intelligent Function Module

APPENDIX
- Appendix-1 Reading and Writing by Designation of the Device Memory Extension
- Appendix 2 Reading from and Writing to the Buffer Memory
- Appendix-3 Processing Time of the PLC CPU Side While Communicating Using the MC Protocol

A - 15

About Manuals

The following manuals are also related to this product.
In necessary, order them by quoting the details in the tables below.

Related Manuals

Manual name	Manual number (Model code)
Q Corresponding Ethernet Interface Module User's Manual (Application) This manual explains the e-mail function of the Ethernet module, the PLC CPU status monitoring function, the communication function via MELSECNET/H, MELSECNET/10 as well as the communication function using data link instructions, and how to use file transfer (FTP server), etc. (sold separately)	SH-080010 (13JL89)
Q Corresponding Ethernet Interface Module User's Manual (Web function) This manual explains how to use the Web function of the Ethernet module. (sold separately)	SH-080180 (13JR40)
Q Corresponding MELSEC Communication Protocol Reference Manual This manual explains the communication methods and control procedures through the MC protocol for the external devices to read and write data from/to the PLC CPU using the serial communication module/Ethernet module. (sold separately)	SH-080008 (13JF89)

Conformation to the EMC Directive and Low Voltage Instruction

For details on making Mitsubishi PLC conform to the EMC directive and low voltage instruction when installing it in your product, please see Chapter 3, "EMC Directive and Low Voltage Instruction" of the User's Manual (Hardware) of the CPU module to use.
The CE logo is printed on the rating plate on the main body of the PLC that conforms to the EMC directive and low voltage instruction.
For information about conforming this product to the EMC directive and low voltage instruction , please refer to Chapter 3 "EMC Directive and Low Voltage Instruction", section "3.1.3. Cable" of the User's Manual (Hardware) for the CPU module to use.

A - 16

● How to use this manual

In this manual, explanations are given for each application of the Ethernet modules (QJ71E71-100, QJ71E71-B5 and QJ71E71-B2).
Please use this manual using the following key items below as a reference.

(1) To find out about the features and utility lists

(a) To find out about the features and functions
- Chapter 1 describes the features of the Ethernet modules.
- Chapter 3 describes the common functions and specifications of the Ethernet modules.

(b) To find out about the packaged items and network configured items
- The section prior to Chapter 1, "Product Configuration" describes the items that are supplied together with the Ethernet module.
- Section 2.2 describes the system configuration of the Ethernet module. Parts and components other than those packaged with the module must be purchased separately by the user.

(2) To find out about the processing required prior to starting the operation of the Ethernet module

(a) To find out about the startup procedure
- Section 4.2 describes an outline of the procedures prior to starting the operation of the Ethernet module.

(b) To find out about the connection to the Ethernet network system.
- Section 2.2 describes the devices required to connect to the Ethernet network system.
- Section 4.4 describes the connection methods for each type of interface.

(c) To find out about the parameter settings required prior to starting the operation of the Ethernet module
- Section 4.5 describes the types of the parameter setting screens for GX Developer in order to use the Ethernet module.
- Section 3.6 describes the parameter settings required for each function to be used.
  Confirm the required parameters, set them according to the relevant section providing detailed explanation, and save the setting values in the PLC CPU to which the Ethernet module is installed.

(d) To find out how to check for Ethernet module failures
- Section 4.8 describes the self-diagnostic test for the Ethernet module.

A - 17

(e) To find out how to check for connection errors with the external devices

- Sections 5.4.1 to 5.4.3 describe how to check for connection errors by performing the PING test and loop back test through GX Developer.
- Section 5.4.4 describes how to check for connection errors using the "PING" command.
- Section 5.4.5 describes how to check for connection errors by performing the loopback test through MC protocol-based communication.

  ✻ For details of the loopback test commands through MC protocol, refer to the Q Corresponding MELSEC Communication Protocol Reference Manual.

(3) To find out about the connection between the Ethernet module and the external devices

(a) To find out about the communication procedures

- Section 5.1 describes an outline of the communication procedures

(b) To find out about the connections with the external devices

- Section 5.6 describes the connections (open and close processing) for each of the communication method (TCP/IP, UDP/IP) and the open method (Active, Passive), including programming procedures.

(4) To find out about the details of the data communication functions

(a) To find out about the communication functions

- Section 1.2 describes an overview of the Ethernet module communication functions and related section numbers and manual names that can be referenced for more detailed explanations.
- Special functions of the Ethernet module are described in the User's Manual (Application).
- Web functions of the Ethernet module are described in the User's Manual (Web function).

(5) To find out about the data communication functions and programming

(a) To find out how data is read from and written to the PLC CPU

- Data is read from and written to the PLC CPU with communication functions using the MC protocol.
- Chapter 6 describes an overview of the communication functions using the MC protocol.

  ✻ For details, refer to the Q Corresponding MELSEC Communication Protocol Reference Manual.

(b) To find out how to send and receive data between the PLC CPU and the external devices

- Data communication between the PLC CPU and the external devices is performed with the communication functions using either the fixed buffers or the random access buffers.
- Chapters 7 and 8 explains details of the communication functions and programming using the fixed buffers.
- Chapter 9 explains details of the communication functions and programming using the random access buffers.

(6) To find out how to check for error occurrences and take corrective actions

(a) To find out about the contents of the error codes

- Chapter 11 describes troubleshooting, how to check for errors, and contents and reference manual of error codes.

(b) To find out about the storage area of the error codes in the buffer memory of the Ethernet module

- Section 11.3 describes the error code storage areas in the buffer memory.

(7) To learn about functions that have been added to function version B and later

- Section 1.3 provides the added function list and detailed explanation manual.
- Appendix 1.1 provides the comparison of the Ethernet module functions.

● Structure of this manual

(1) Setting parameters with GX Developer

(a) Using GX Developer to set parameters, the sequence programs for communicating with external devices can be simplified in the Ethernet module.

(b) In this manual, parameter settings using GX Developer are explained as follows.

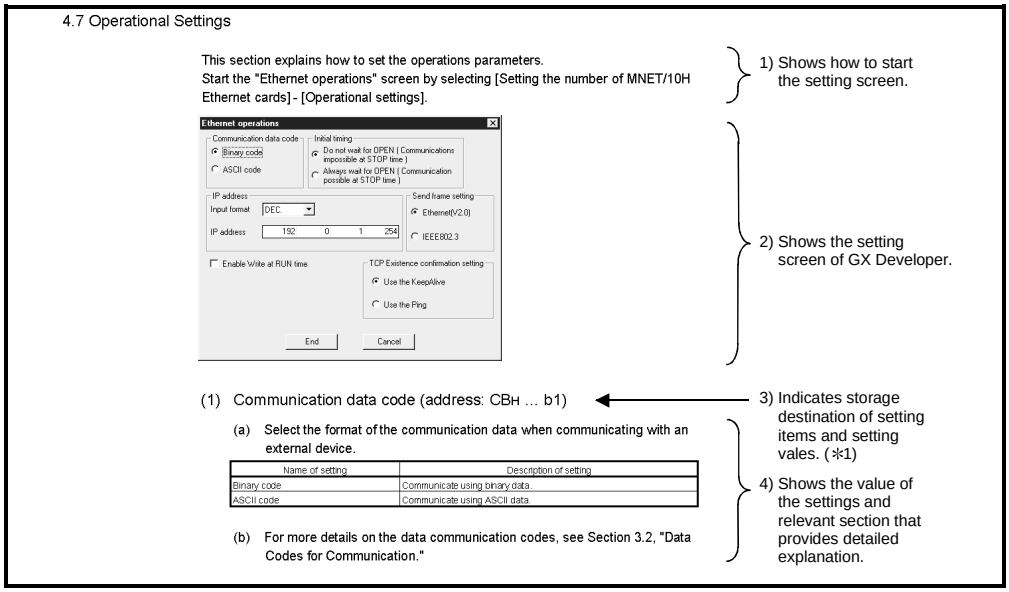
1) Section 4.5 describes the types of setting screens, objectives of settings, items and an outline of settings.

2) Details are explained in the detailed explanatory sections on the screens or in the text in Section 4.5.

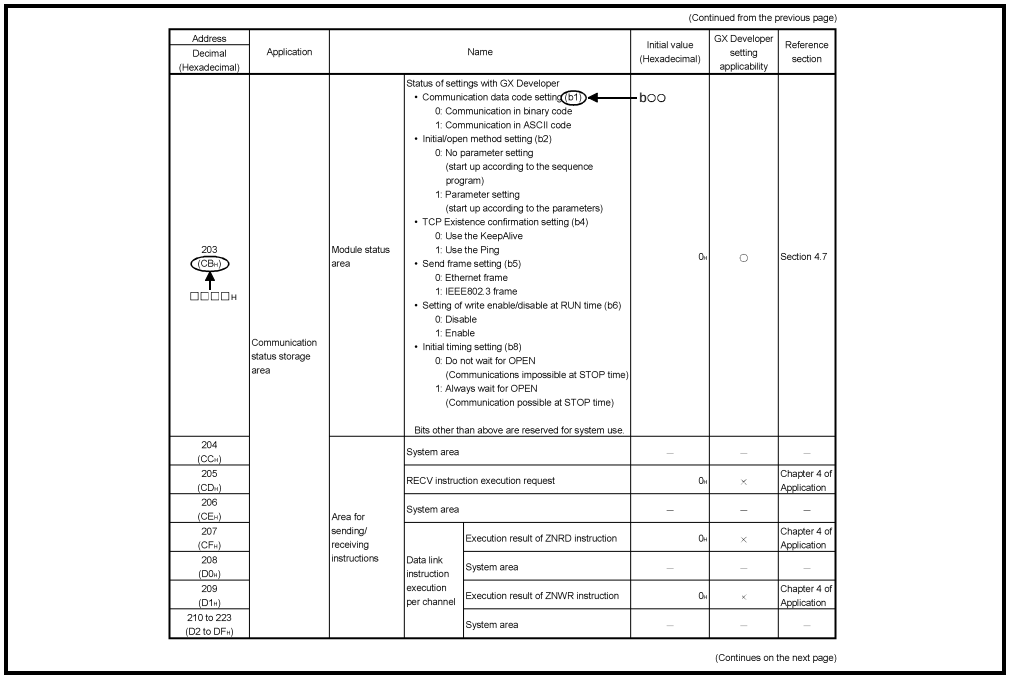
(c) Confirm the required parameters and set them according to the relevant explanatory section or manual, and save the values in the PLC CPU to which the Ethernet module is installed.

(2) Setting screen of GX Developer

This manual explains the parameter settings with GX Developer in the following format:


4.7 Operational Settings
This section explains how to set the operations parameters.
Start the "Ethernet operations" screen by selecting [Setting the number of MNET/10H Ethernet cards] - [Operational settings].
1) Shows how to start the setting screen.
Ethernet operations
Communication data code
Binary code
ASCII code
Initial timing
Do not wait for OPEN (Communications impossible at STOP time)
Always wait for OPEN (Communication possible at STOP time)
IP address
Input format
DEC.
IP address
192 0 1 254
Send frame setting
Ethernet(V2.0)
IEEE802.3
Enable Write at RUN time
TCP Existence confirmation setting
Use the KeepAlive
Use the Ping
End
Cancel
2) Shows the setting screen of GX Developer.
(1) Communication data code (address: CBH ... b1)
3) Indicates storage destination of setting items and setting vales. (*1)
(a) Select the format of the communication data when communicating with an external device.
Name of setting
Description of setting
Binary code
Communicate using binary data.
ASCII code
Communicate using ASCII data.
(b) For more details on the data communication codes, see Section 3.2, "Data Codes for Communication."
4) Shows the value of the settings and relevant section that provides detailed explanation.

* The value in parentheses (address: □□□□H ... b○○) indicates the buffer memory address and the bit position of the Ethernet module that store a setting value entered from GX Developer.
For details on the buffer memory, see Section 3.8, "List of Applications and Assignments of the Buffer Memory."



A - 20

About the Generic Terms and Abbreviations

This manual uses the following generic terms and abbreviations to describe the Model QJ71E71-100, QJ71E71-B5 and QJ71E71-B2 Ethernet interface modules, unless otherwise specified.

Generic Term/Abbreviation	Description	
ACPU	Generic term for AnNCPU, AnACPU, and AnUCPU	
AnACPU	Generic term A2ACPU, A2ACPU-S1, A2ACPUP21/R21, A2ACPUP21/R21-S1, A3ACPU, A3ACPUP21/R21	
AnNCPU	Generic term A1NCPU, A1NCPUP21/R21, A2NCPU, A2NCPU-S1, A2NCPUP21/R21, A2NCPUP21/R21-S1, A3NCPU, A3NCPUP21/R21	
AnUCPU	Generic term for A2UCPU, A2UCPU-S1, A2ASCPU, A2ASCPU-S1, A3UCPU, and A4UCPU	
Ethernet Address	A machine-specific address that is also referred to as the MAC (Media Access Control) address. This is used to identify the addresses of external devices over a network. The Ethernet address of the Ethernet module can be verified on the MAC ADD column of the rating plate.	
Ethernet module	Abbreviation for Model QJ71E71-100, QJ71E71-B5 and QJ71E71-B2 Ethernet Interface Modules (Described as the Ethernet module or E71 in the figures)	
Ethernet network system	Abbreviation for 10BASE2,10BASE5, 10BASE-T and 100BASE-TX network systems	
External device	Generic term for personal computers, computers, workstations (WS) and Ethernet module etc. that are connected by the Ethernet for data communication	
GX Developer	Abbreviation for GX Developer (SWnD5C-GPPW-E). (n in the model name is 4 or greater)	
MELSECNET/10	Abbreviation for MELSECNET/10 Network system	
MELSECNET/H	Abbreviation for MELSECNET/H Network system	
MX Component	Abbreviation for MX Component (SW0D5C-ACT-E or later)	
Network module (N/W module)	Abbreviation for interface modules compatible with the MELSECNET/10 (H) network system	
OPS	Generic term for the partner product installed with redundant system-compatible EZSocket (operator station). The OPS can make communication using the Ethernet module's user connection for OPS connection. (Refer to Section 5.5.)	
Personal computer	Generic term for IBM PC/AT (or 100% compatible) personal computer	
QCPU	Q mode	Generic term for Q00JCPU, Q00CPU, Q01CPU, Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, and Q25PRHCPU
QCPU-A	A mode	Generic term for for Q02CPU-A, Q02HCPU-A, and Q06HCPU-A
QCPU station	Abbreviation for the PLC with QCPU installed.	
QnACPU	Generic term for Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU, and Q4ARCPU	
Q/QnA	Generic term for QCPU and QnACPU	
Reference Manual	Abbreviation for the Q Corresponding MELSEC Communication Protocol Reference Manual	
User's Manual (Application)	Abbreviation for the Q Corresponding Ethernet Interface Module User's Manual (Application)	
User's Manual (Basic)	Abbreviation for the Q Corresponding Ethernet Interface Module User's Manual (Basic)	
User's Manual (Web function)	Abbreviation for the Q Corresponding Ethernet Interface Module User's Manual (Web function)	

Product Configuration

The following lists the product configuration of the Ethernet interface modules.

Model	Item name	Quantity
QJ71E71-100	QJ71E71-100 Ethernet interface module	1
QJ71E71-B5	QJ71E71-B5 Ethernet interface module	1
QJ71E71-B2	QJ71E71-B2 Ethernet interface module	1

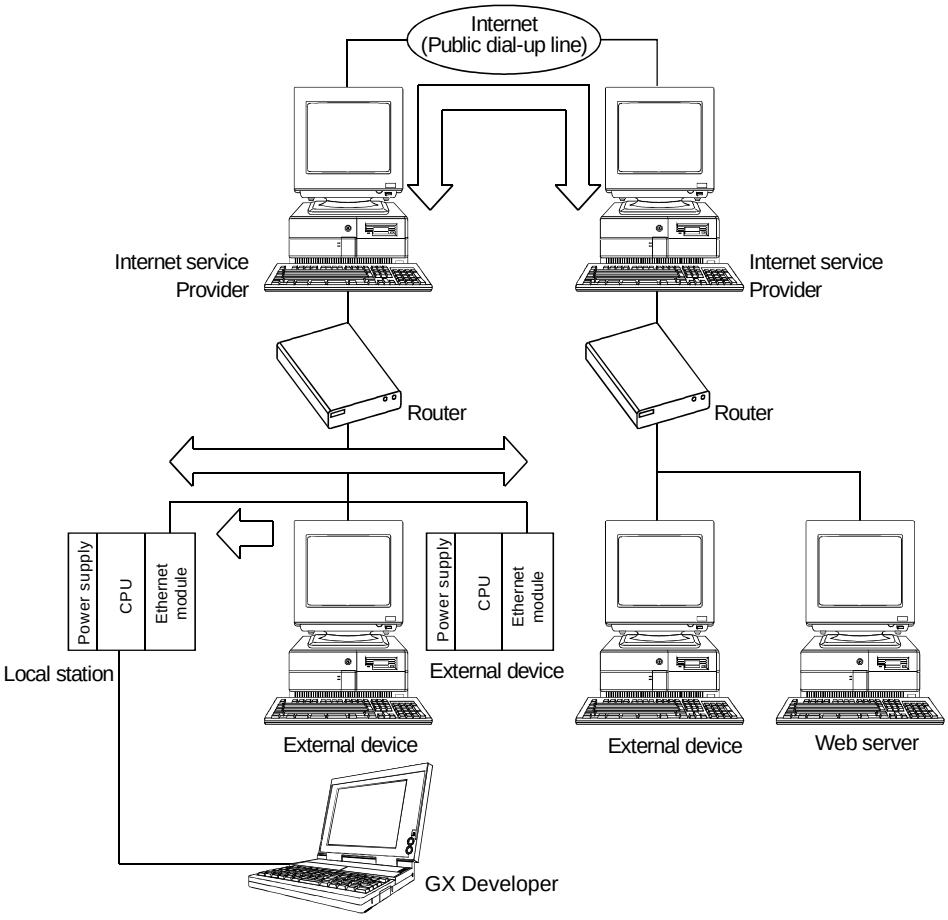
A - 22

1 OVERVIEW

This manual provides information on the specifications of the Ethernet interface modules, models QJ71E71-100, QJ71E71-B5 and QJ71E71-B2 (hereinafter called the Ethernet module), as well as the procedures prior to starting the operation, the control procedures and data communication method for communicating with external devices, maintenance, inspection, and troubleshooting.
When applying the following program examples to the actual system, make sure to examine the applicability and confirm that it will not cause system control problems.

1.1 Overview of the Ethernet Module

The Ethernet module is an interface module on the PLC side for connecting the Q series PLC with the host system, such as a personal computer and a work station, and between PLCs using the TCP/IP or UDP/IP communication protocol via Ethernet (100BASE-TX, 10BASE-T, 10BASE5, 10BASE2).



1) Collection and modification of PLC CPU data
(Communication using the MELSEC communication protocol)
2) Transmission and reception of arbitrary data to/from external devices
(Communication using fixed buffers or random access buffers)
3) Data transmission/reception by e-mails
(When using the e-mail function)
4) Data transmission/reception by Web function
(User's Manual (Web function))

* By using the GX Developer (SW4D5C-GPPW-E or later version, hereinafter called GX Developer), the sequence programs for communication can significantly be simplified.

1.2 Features of the Ethernet Module

1

(1) Data communication using the MELSEC communication protocol (Details are explained in Chapter 6 of the MELSEC Communication Protocol Reference Manual)

In the "data communication using the MELSEC communication protocol (hereinafter called the MC protocol)", the device data and program files of the PLC can be read from/written to the host system.

This protocol is a passive protocol that communicates data solely according to the requests from the host system. It does not require a sequence program for data communication after a connection is established.

If the host system is a personal computer running one of the basic operation systems below, it is possible to create a communication program for the host system without considering the detailed MC protocol (transmission/reception procedures) using one of the following separately sold communication support tools.

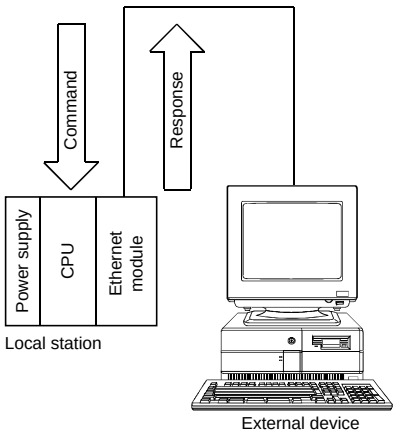
(Supported basic operation systems)

- Microsoft® Windows® 95 Operating System
- Microsoft® Windows® 98 Operating System
- Microsoft® Windows NT® Workstation Operating System Version 4.0
- Microsoft® Windows® Millennium Edition Operating System
- Microsoft® Windows® 2000 Professional Operating System
- Microsoft® Windows® XP Professional Operating System
- Microsoft® Windows® XP Home Edition Operating System

＊ Depending on the version of MX Component used, different operating systems are supported.
See the manual of MX Component for the details.

(Separately sold communication support tools)

- MX Component (SW0D5C-ACT-E or later, hereinafter abbreviated as MX Component)
  ＊ See Appendix 9 for the overview of MX Component.



REMARK

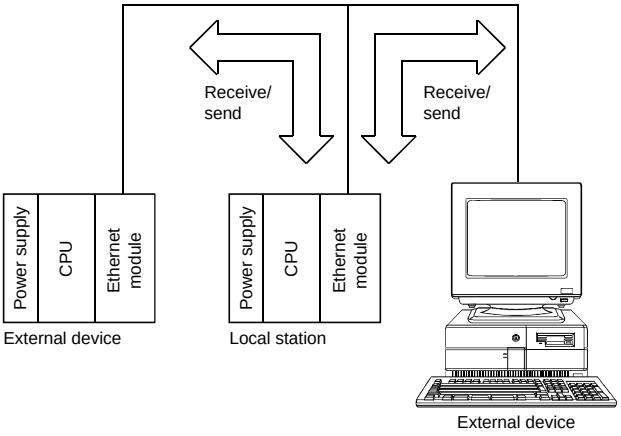
The communication functions using the MC protocol correspond to the communication functions for reading/writing data from/to the PLC CPU that are supported by the A/QnA series Ethernet modules (A1SJ71E71/A1SJ71QE71, etc.).

(2) Communication using fixed buffers (Details are explained in Chapters 7 and 8)

In the "communication using fixed buffers," a maximum of 1 k words of arbitrary data can be sent or received among PLCs or between the PLC and the host system.

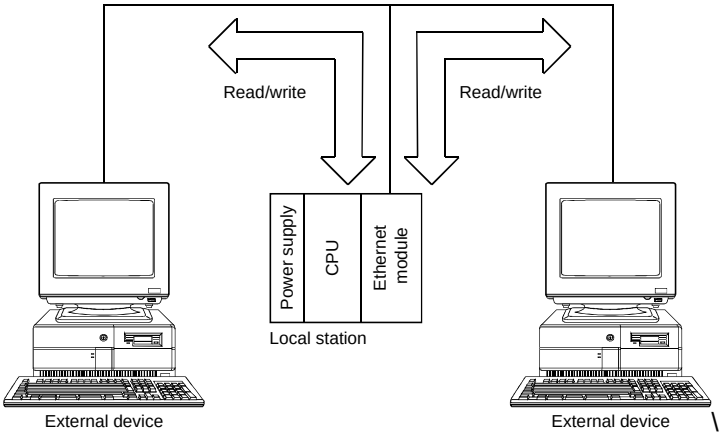
An Ethernet module is provided with 16 fixed buffer data areas of 1 k word storage space, and each is assigned as either a sending or receiving buffer for an arbitrary device.

While the communication using the MC protocol is passive, the communication using fixed buffers is an active protocol. Data can be sent from the PLC side to the host system when errors occur in machine equipment or when some conditions are satisfied. Furthermore, by using the data receiving function in an interrupt program, retrieval of receive data to the PLC CPU may be expedited.



(3) Communication using random access buffers (Details are explained in Chapter 9)

In the "communication using random access buffers," data of larger size (up to 6 k words of data) can be communicated. This protocol can be used when the data size is too large for communication using fixed buffers (up to 1 k words of data).



(4) Communication by e-mails (Details are explained in the User's Manual (Application))

With "sending/receiving e-mail," data can be sent to and received from an external device at a remote location using e-mail via an Internet line.

(a) Sending/receiving e-mail by the PLC CPU

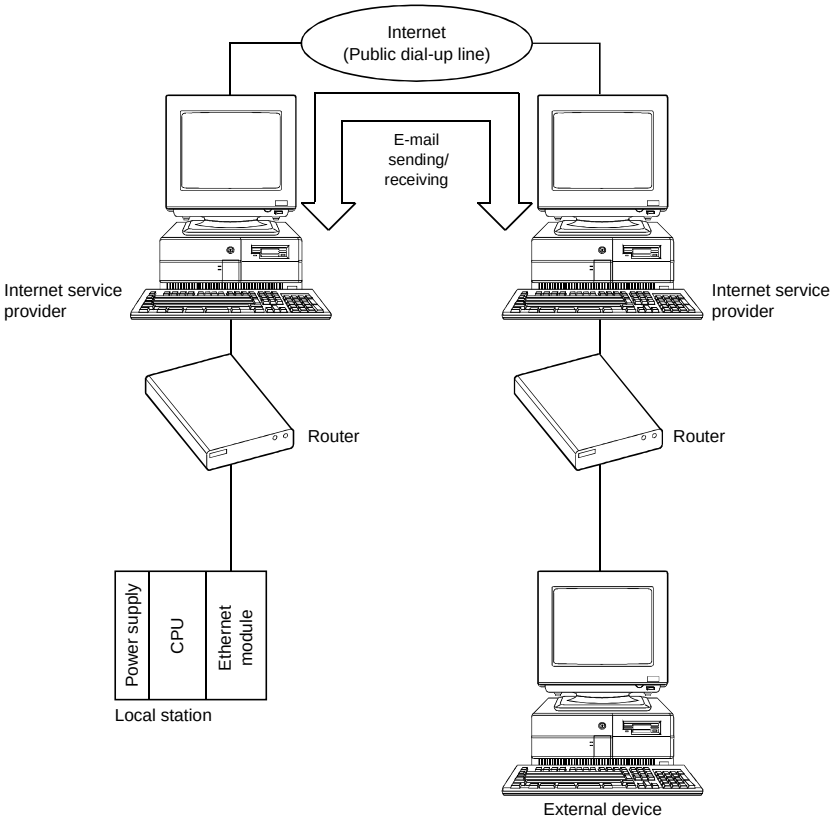
The following data can be sent/received by using dedicated instructions (MSEND, MRECV).

1) Sending/receiving data as attached files
   Up to 6K words of data can be sent to or received from a personal computer or other Ethernet module as a file attached to e-mail.
2) Sending data as main text
   Up to 960 words of data can be sent to a personal computer or portable terminal as main text of e-mail.

(b) Sending e-mail by the PLC CPU Monitoring function

By setting up the Ethernet parameters, the notification conditions (PLC CPU status or device value) set by the user can be monitored at constant intervals, and up to 960 words of data can be sent by either of the following methods when the notification conditions are satisfied:

1) Sending data as an attached file
2) Sending data as main text



(5) Communication by the Web function (Details are explained in the User's Manual (Web function))

With "communication using the Web function," the system administrator can monitor a Q Series CPU at a remote location via the Internet using a commercially available Web browser.

(a) By setting up a communication library in the Web server, data communication with the PLC can be performed.

A sample screen to be displayed by a Web browser is also available for your use.

Contact your local agency or marketing company.

(b) A Web server and a Web browser are required to use the Web function.

(Basic operating systems)

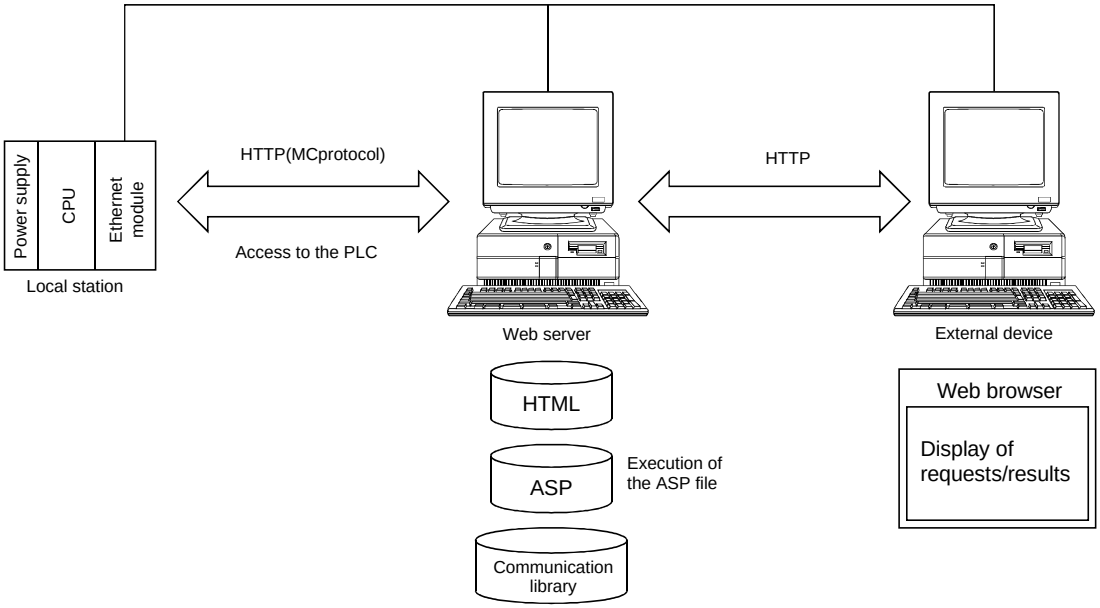
- Microsoft® Windows® 2000 Server Operating System
- Microsoft® Windows® 2000 Professional Operating System
- Microsoft® Windows NT® Server Network Operating System Version 4.0
- Microsoft® Windows NT® Workstation Operating System Version 4.0
- Microsoft® Windows® 98 Operating System

(Web servers)

- Microsoft® Internet Information Server 5.0
- Microsoft® Internet Information Server 4.0
- Microsoft® Peer Web Services 4.0
- Microsoft® Personal Web Server 4.0

(Web browsers)

- Internet Explorer 4.0 or later (Microsoft® Corporation)
- Netscape® Communicator 4.05 or later (Netscape® Communications Corporation)



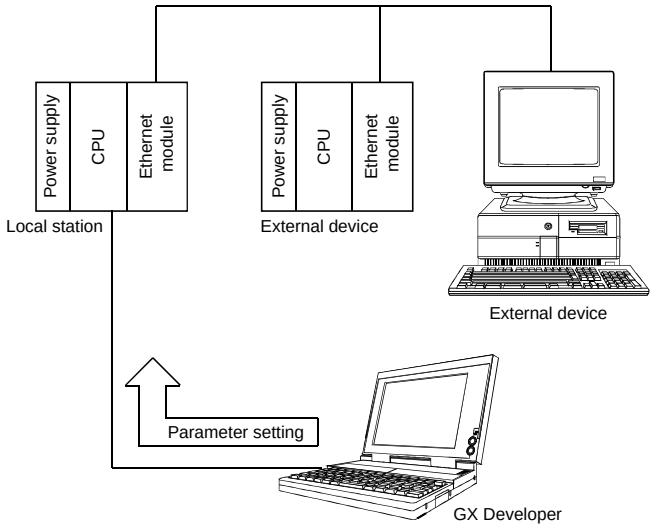
(6) Connecting GX Developer, GT SoftGOT and GOT (Details are explained in the Operating Manuals for GX Developer and GT SoftGOT, and User's Manual for GOT (Connection System Manual))

(a) Simplifying sequence programs using GX Developer

GX Developer supports the parameter setting function to perform the Ethernet module initialization and the open processing with external devices. By setting up the following parameters with "network parameter settings" of GX Developer, access from the external device to the PLC is enabled. It can also significantly simplify sequence programs used to perform communication by Ethernet modules.

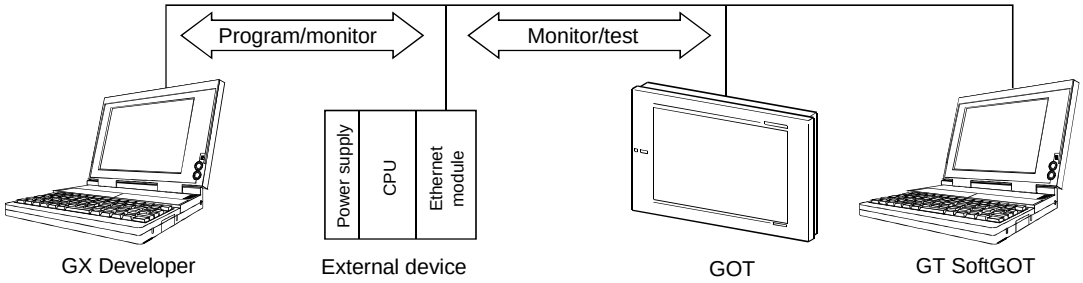
- IP address setting
- Port number setting
- Protocol type setting
- Notification condition setting

For more information on setting up Ethernet modules, see Section 3.6, "List of GX Developer Setting Items for Ethernet Modules" and other applicable reference sections.



(b) Programming and monitoring function via the Ethernet

By establishing an Ethernet connection, programming and monitoring of PLCs can be performed using GX Developer, as well as monitoring and testing of PLCs can be performed using GOT or GT SoftGOT.
In ether case, remote operations utilizing long-distance connection and high-speed communication of Ethernet become possible.



(c) Connecting multiple MELSOFT products (GX Developer, GT SoftGOT, and MX Component) or GOTs

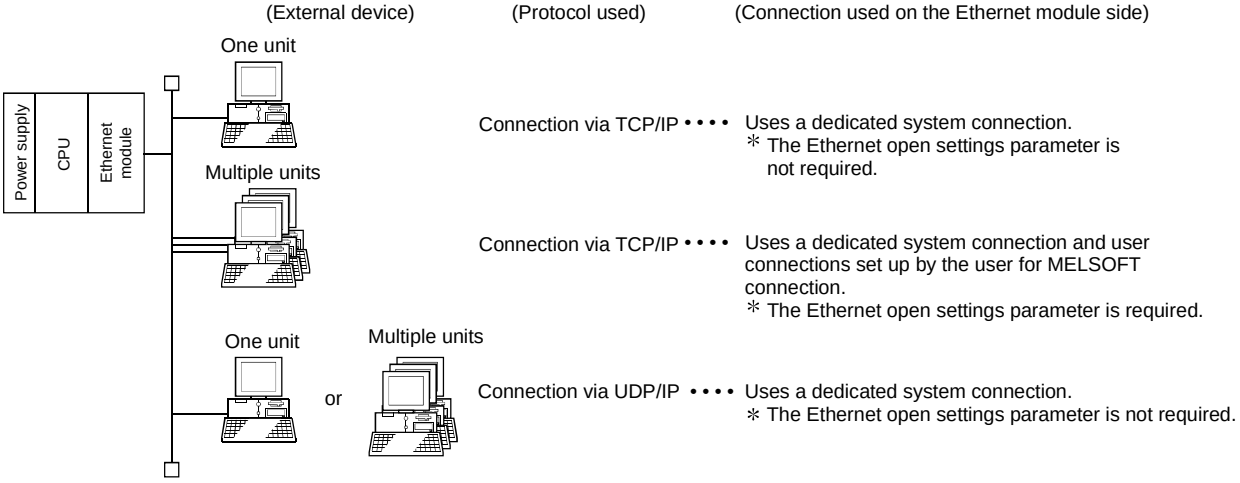
This product can be connected with one or more MELSOFT product (GX Developer, GT SoftGOT, and MX Component) or GOT simultaneously via TCP/IP communication or UDP/IP communication. ∗1 ∗2

1) Connection via TCP/IP communication
   - The Ethernet module side can connect up to 17 units of MELSOFT product via TCP/IP communication simultaneously by using one dedicated system connection and up to 16 user connections. ∗3
   - If only one MELSOFT product is to be connected, the following settings using GX Developer are not required.
     If two or more MELSOFT product are to be connected, the following settings using GX Developer are required since user connections will be used.
     On the "Ethernet open settings" screen for network parameters, set "TCP" in the protocol field of the connection number to be used, and "MELSOFT connection" in the open method field. (See Section 5.5.)

2) Connection via UDP/IP communication. ∗4

   By using one dedicated system connection, the Ethernet module side can connect MELSOFT product or GOT via UDP/IP communication.

[Connections used by the Ethernet module side when connecting a MELSOFT product or GOT]


(External device)
(Protocol used)
(Connection used on the Ethernet module side)
Power supply
CPU
Ethernet module
One unit
Multiple units
One unit
or
Multiple units
Connection via TCP/IP • • • •
Uses a dedicated system connection.
∗ The Ethernet open settings parameter is not required.
Connection via TCP/IP • • • •
Uses a dedicated system connection and user connections set up by the user for MELSOFT connection.
∗ The Ethernet open settings parameter is required.
Connection via UDP/IP • • • •
Uses a dedicated system connection.
∗ The Ethernet open settings parameter is not required.

∗1 It is possible to use the same station number if two or more MELSOFT products are started up from one personal computer in order to perform TCP/IP communication and UDP/IP communication with one Ethernet module. (It is not necessary to use different station numbers for TCP/IP and UDP/IP.)

∗2 GT SoftGOT and GOT support UDP/IP communication only.

∗3 Dedicated connections will be used to connect MELSOFT products for data communication. These dedicated connections cannot be used to perform data communication with external devices other than MELSOFT product.

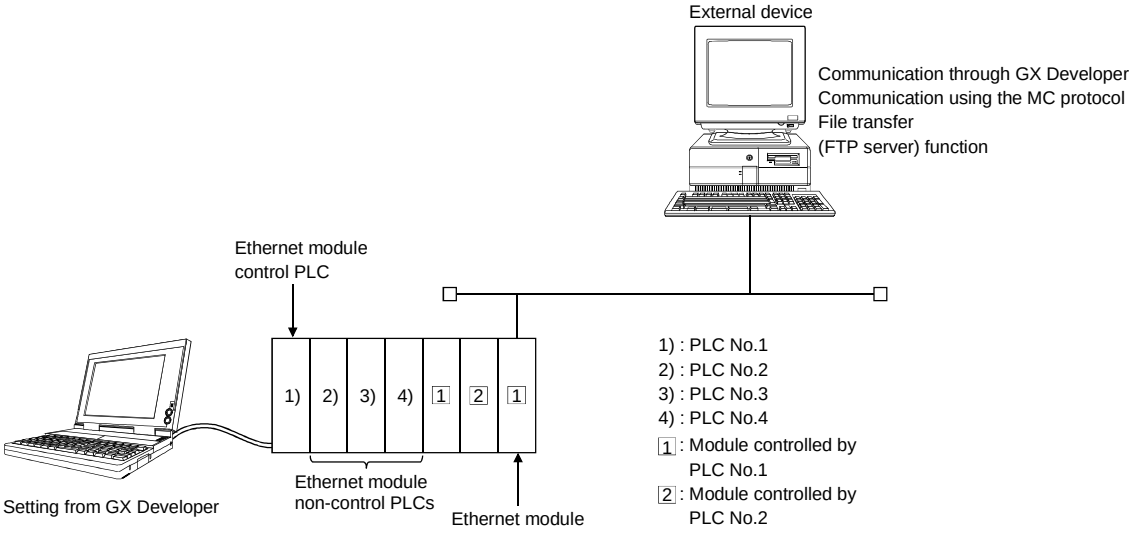
∗4 It is possible to access other stations without performing Conversion setting (Network No., Station No., and IP address) if they are accessed from a MELSOFT product via the Ethernet module using the Communication with MELSECNET/H or MELSECNET/10 relay.
See Chapter 3 of User's Manual (Application) for the relay function of MELSECNET/H and MELSECNET/10.

(7) Functions supporting Multiple CPU systems (Details are explained in the Reference Manual.)

(a) When performing the following data communication with QCPUs in a multiple CPU system, it is possible to perform data communication such as reading/writing device data or reading/writing files by specifying the QCPU to be accessed.

1) Communication by MC protocol
2) Communication by GX Developer
3) When using the file transfer function (FTP server)

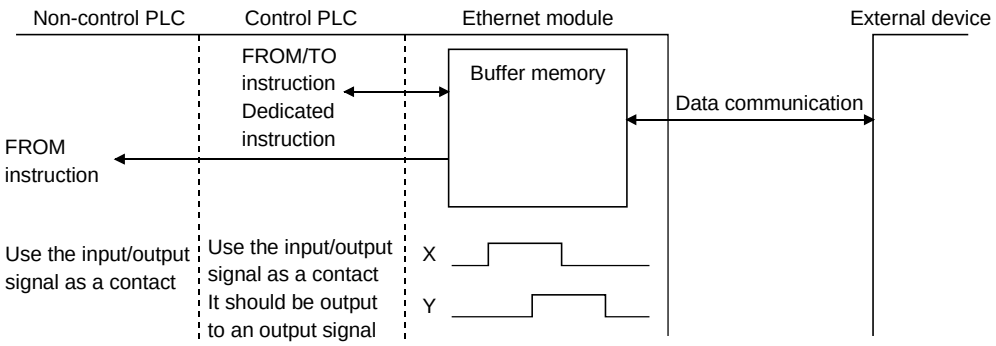
* When using the Ethernet module in a multiple CPU system, a QCPU controlling the Ethernet module (hereinafter referred to as the control PLC) should be specified through GX Developer.
It is also possible to mount an Ethernet module of function version A on a multiple CPU system and access to the only control PLC (the PLC No.1).



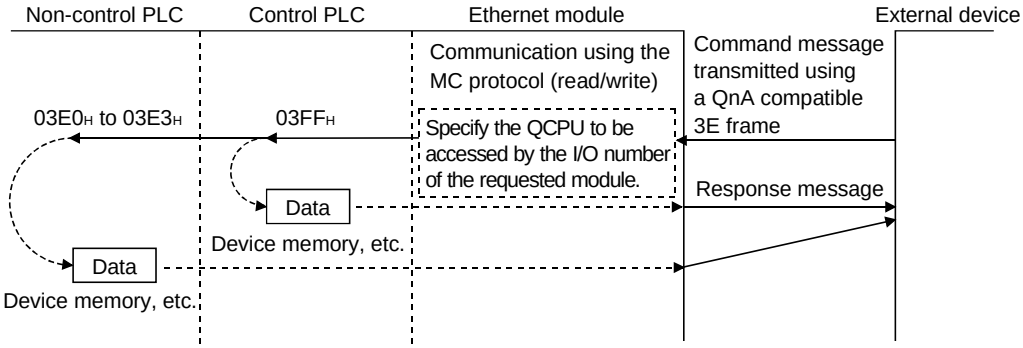
(b) When the Ethernet module of function version B or later is used in a multiple CPU system, the following data can be transferred to/from the Ethernet module.

1) Communication using the fixed buffer, communication using data link instructions and sending/receiving of e-mail are possible from the control PLC.

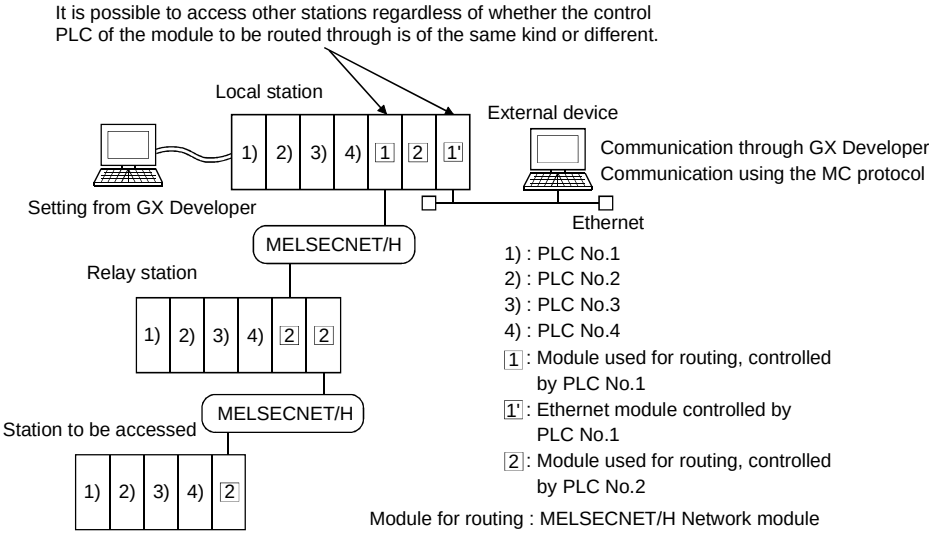
2) It is possible to read the buffer memory from non-control PLCs. Input/output signals can be used as contacts.



3) It is possible to access the control PLC and non-control PLCs using the MC protocol and through GX Developer and file transfer (FTP server) from the external device.
Also, communication using the fixed buffer and sending/receiving of e-mail are possible with respect to the Ethernet module control PLC.
(Example)
When communicating by MC protocol



If the MC protocol or GX Developer is used to access other stations, it is possible to access the control PLC and non-control PLCs of the station to be accessed even if the relay station and the accessed station are multiple CPU systems.
(Example)


It is possible to access other stations regardless of whether the control PLC of the module to be routed through is of the same kind or different.
Local station
External device
1) 2) 3) 4) 1 2 1'
Communication through GX Developer
Communication using the MC protocol
Setting from GX Developer
Ethernet
MELSECNET/H
Relay station
1) 2) 3) 4) 2 2
MELSECNET/H
Station to be accessed
1) 2) 3) 4) 2
1) : PLC No.1
2) : PLC No.2
3) : PLC No.3
4) : PLC No.4
1 : Module used for routing, controlled by PLC No.1
1' : Ethernet module controlled by PLC No.1
2 : Module used for routing, controlled by PLC No.2
Module for routing : MELSECNET/H Network module

* The QnA compatible 3E frame should be used for access to non-control PLCs when communicating using the MC protocol.
Note, however, that the available functions differ, depending on the QCPU to be accessed (whether it is a control PLC or a non-control PLC).
See the Reference Manual for the available functions and accessible range.
* The modules used for routing are indicated the following when accessing other stations:

- MELSECNET/H, MELSECNET/10 network module
- Q series C24
- Ethernet module

* If there is a module of function version A among the modules for routing, it is possible to access the control PLC of module used for routing only. In addition, it is possible to access other stations via a module controlled by the same control PLC.

(8) Remote password check function (Details are explained in Chapter 5 of the Reference Manual.)

(a) The remote password check function of the Ethernet module prevents improper access to the QCPU by users at a remote location.
The Ethernet module performs a remote password check with respect to data communication from the external device that uses the connection set as a parameter in the QCPU.

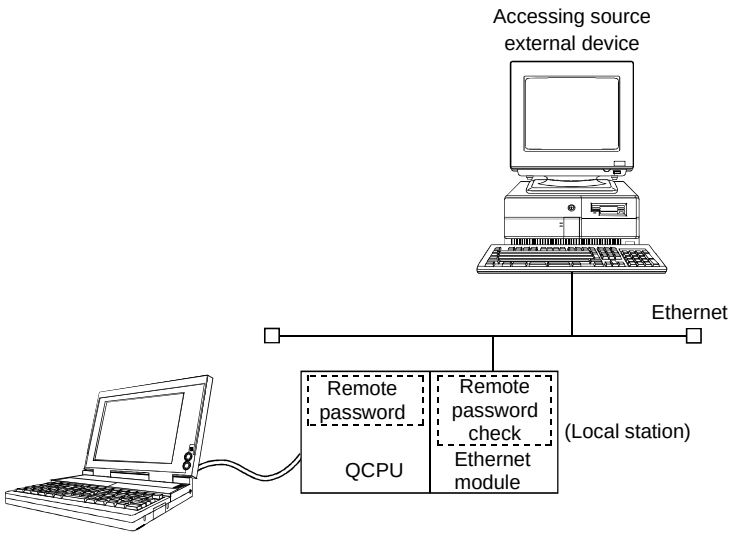
* The remote password function is one of the QCPU functions for preventing improper access to the QCPU by other users.
When setting a remote password for the QCPU using GX Developer the remote password function of the QCPU can be used.

(b) All the remote password checks during data communication from the external device to the local station or other stations are performed for the remote passwords set in the local station QCPU.

(c) When communicating data using the applicable connection for the remote password check, data communication from external device can be performed by the remote password unlocked (canceled) processing after completing open processing.

1) When communicating using the MC protocol, fixed buffer or random access buffer
Use the dedicated instruction for communication using the MC protocol to unlock the remote password from the external device.
2) When accessing the PLC through GX Developer
Unlock the remote password through GX Developer at the start of online operation.
3) When using the file transfer (FTP server) function
Use a dedicated FTP command to unlock the remote password from the external device.
4) When using the Web function
Unlock the remote password in the dialog box displayed by a Web browser while accessing the QCPU.

The user performs remote password lock processing prior to closing the connection.



Setting from GX Developer
Set the applicable connection for the remote password check with the parameters.

(9) Redundant system support function (Detailed explanation: Chapter 5)

(a) Backup of Ethernet modules

By mounting the Ethernet modules of function version D or later on the main base units of a redundant system, the Ethernet modules can be backed up. If a failure or communication error occurs in the Ethernet module, the system can be switched between the control system and standby system to continue the communication of the Ethernet module. *1

(b) Issue of system switching request to control system CPU

When the Ethernet module mounted on the main base unit of the control system CPU in the redundant system detects a communication error or disconnection, it can issue a system switching request to the control system CPU.

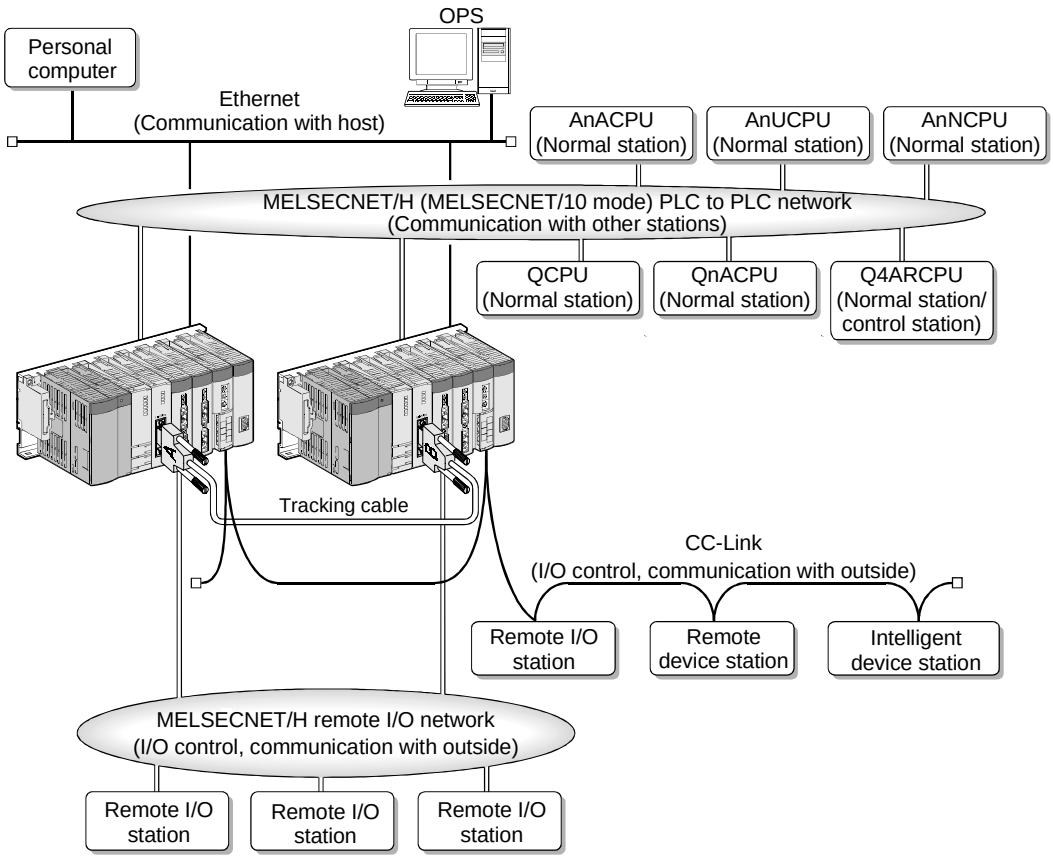
(c) Access to redundant system

Communication using MC protocol or data link instruction enables data communication, such as read/write of device data and files, from/to the control system/standby system or system A/system B of the redundant system.

(d) Connection with OPS

The Ethernet module can communicate with the OPS using the user connection for OPS connection.
Set the user connection for OPS connection in the open setting of GX Developer. (Refer to Section 5.5.)

*1 The module can also be mounted on the MELSECNET/H remote I/O station of the redundant system. Since there are restrictions on the available functions, however, refer to Section 2.6 when mounting the module on the MELSECNET/H remote I/O station.



1.3 Additional Functions in Function Version B or Later

The following table shows the functions that have been added to the Ethernet modules of function version B and later.
Refer to Section 2.7 for the function versions, serial numbers and software versions of the Ethernet modules that can use added functions and of related products (CPU modules, GX Developer).
For function comparison by the function version of Ethernet modules, see Appendix 1.1.

(1) Functions added to function version B

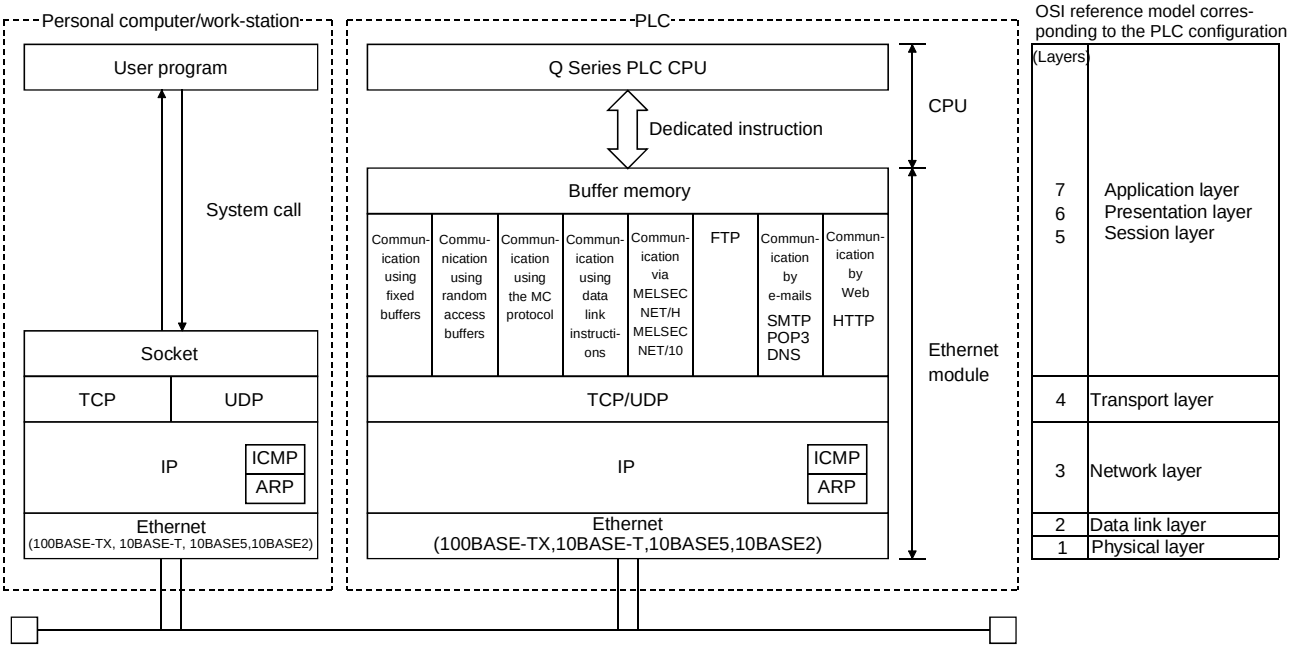
Function		Overview	Reference section
Support for IEEE802.3 frames		Sends/receives data using IEEE802.3 frames. • When sending data, selects and sends either an Ethernet frame or IEEE802.3 frame. • When receiving data, receives either an Ethernet frame or IEEE802.3 frame.	Section 4.7 Appendix 9
Re-initial processing of the Ethernet module		Re-initial processing is performed in the following cases: • When updating the Ethernet address of the external device with which data communication will be performed • When changing the data communication conditions.	Section 5.2.3
Existence check function	Checking by KeepAlive	Perform a target existence check by sending an ACK message to a remote device and waiting to see whether or not a response is received.	Section 5.2.2, REMARK (5)
Simultaneous connection of up to 17 MELSOFT products via TCP/IP communication		Simultaneously connects up to 17 MELSOFT product (such as GX Developer) via TCP/IP communication. This is achieved by using up to 16 user connections and one dedicated system connection.	Sections 1.2 (6) and 5.5 Applicable MELSOFT product manual
Simplifying connection with MELSOFT products	Simplifying access to other stations	It is possible to access other stations without performing conversion setting (network number, station number, and IP address) if they are accessed from a MELSOFT product via the Ethernet module.	Section 1.2 (6) Manual of each MELSOFT product
	Access with the same station number	It is possible to use the same station number if two or more MELSOFT products are started up from one personal computer in order to perform TCP/IP communication and UDP/IP communication with one Ethernet module.	
Ethernet diagnostic function of GX Developer		The Ethernet diagnostic function of GX Developer enables the following diagnoses: • Monitoring the module status and error status of the Ethernet module. • Confirming the completion of the Ethernet module's initial processing by performing the PING test and loop back test.	Sections 5.4 and 11.2
When using the e-mail function	Sending files in CSV format as attachment	Sends a file in CSV format as attachment to e-mail from the Ethernet module.	Chapter 2 of User's Manual (Application)
	Sending main text	Sends main text of up to 960 words from the Ethernet module.	
	Support for encoding/decoding	The following encoding/decoding is supported. • Encode the Subject using 7 bits encoding and send it from the Ethernet module. • Decode and receive an e-mail encoded with Quoted Printable by the Ethernet module.	
Communication using the Web function		Accesses the PLC from a personal computer at a remote location via an Internet, using a commercially available Web browser.	User's Manual (Web Function)
Remote password check	Overview of function	In data communication via the connection set up using the parameters for the QCPU, this function enables data communication after unlock processing from the external device is completed normally for the remote password set in the QCPU.	Section 5.9
	Communication using the MC protocol	Unlocks/locks the remote password of the QCPU. The unlock processing enables access to the QCPU using various commands.	Section 3.18 of Reference Manual
	File transfer (FTP server) function		Section 5.6 of User's Manual (Application)
	Communication using the Web function	Unlock the remote password through the dialog box displayed by a Web browser while accessing the QCPU.	User's Manual (Web Function)
Support for multiple CPU systems	Communication using the MC protocol	This function enables access to the control PLC/non-control PLC specified by the user when performing data communication with a multiple CPU system.	Section 2.10 of Reference Manual
	Accessing QCPU from GX Developer		GX Developer Operating Manual
	File transfer (FTP server) function		Chapter 5 of User's Manual (Application)
Mounting an Ethernet module to the MELSECNET/H remote I/O station		When an Ethernet module is mounted to the MELSECNET/H remote I/O station, access to the local station (station on which the Ethernet module is mounted) or other stations is enabled from the external device.	Section 2.6

(2) Functions added to function version D

Function	Overview	Reference section
Redundant system support function	• The Ethernet modules can be mounted and used on the main base units of the redundant system. (Redundant Ethernet modules) • When a communication error or disconnection is detected, a system switching request can be issued to the control system CPU.	Section 5.11
Hub connection status monitor function	In the buffer memory, the connection status of the Ethernet module and hub and the current transmission speed can be checked.	Section 5.10
Target station CPU type specification for data link instructions	Target station CPU type ((S1)+3) can be specified with the following data link instructions. (The control system/standby system or System A/System B of the redundant system can be specified.) • READ/SREAD instruction • WRITE/SWRITE instruction • REQ instruction	Chapter 4, Chapter 6 of User's Manual (Application)

1.4 Software Configuration

The Ethernet modules support the TCP/IP and UDP/IP protocols.



(1) TCP (Transmission Control Protocol)

This protocol guarantees data credibility or reliability in a communication between a personal computer/work station and the PLC that are connected via network, and provides the following functions:

- Creates a logical connection by establishing a connection (logical line) as if a dedicated line was created between external devices.
- A maximum of 16 connections can be established and communication to multiple buffers can be performed at the same time in the Ethernet module.
- Data reliability is maintained by the sequence control using the sequence numbers, the data retransmission function and the check sum.
- The communication data flow can be controlled by operations using Windows.
- Supports the Maximum Segment option.
  The Maximum Segment option can be made valid for TCP transmission or TCP re-transmission.
  The data reception side must check the length of received data before processing the message.

(2) UDP (User Datagram Protocol)

This protocol may not guarantee data credibility and reliability in a communication between a personal computer/work station and the PLC that are connected via network. Thus, even if the data does not reach the target node, it will not be retransmitted.

- Because it is connectionless, high-speed transmission is possible.
- A check sum is used to increase the reliability of the communication data. However, when greater reliability must be maintained, a user application or the TCP should be utilized.

(3) IP (Internet Protocol)

- Communication data is sent and received in datagram format.
- Communication data can be divided and reassembled.
- Routing option is not supported.

(4) ARP (Address Resolution Protocol)

- This protocol is used to get the Ethernet physical addresses from the IP addresses.

(5) ICMP (Internet Control Message Protocol)

- This protocol is used to exchange errors occurred on an IP network and various information related to the network.
- Provides a function to transmit IP error messages.
- See Appendix for information regarding the option support type (ICMP protocol).

(6) FTP (File Transfer Protocol)

- This protocol is used to transfer files.
- Can upload and download PLC CPU files.

(7) DNS (Domain Name System)

- This system translates IP addresses to names that are easy to remember by the user.

(8) SMTP (Simple Mail Transfer Protocol)

- This protocol transfers mails.

(9) POP3 (Post Office Protocol Ver. 3)

- This protocol transfers mails received by a mail server to a local computer.

(10) HTTP (Hyper Text Transfer Protcol)

- This protocol is used to perform data communication for world wide web of the internet.

2 SYSTEM CONFIGURATIONS

This section explains the system configurations that may be combined with the Ethernet modules.

2.1 Applicable Systems

Ethernet modules can be used with the following systems:

(1) Applicable modules and number of modules that can be mounted

The following table lists the CPU module and network module (for remote I/O station) where the Ethernet module can be mounted and the number of modules that can be mounted.

Applicable module		Number of modules that can be mounted	Remark
CPU module	Q00JCPU Q00CPU Q01CPU	Maximum 1	(*1)
	Q02CPU Q02HCPU Q06HCPU Q12HCPU Q25HCPU	Maximum 4	Can be mounted only in Q mode. (*1)
	Q12PHCPU Q25PHCPU		
	Q12PRHCPU Q25PRHCPU		(*1) (*4)
Network module	QJ72LP25-25 QJ72LP25GE QJ72BR15	Maximum 4	MELSECNET/H remote I/O station (*2) (*3)

*1 Refer to the User's Manual (Function Explanations, Program Fundamentals) of the CPU module to be used.
*2 Refer to the Q Corresponding MELSECNET/H Network System Reference Manual (Remote I/O network).
*3 Use the Ethernet module of function version B or later.
*4 Use the Ethernet module of function version D or later.

(2) Applicable base unit

The Ethernet module can be mounted in any I/O slot (*1) of the base unit.

*1 Limited to within the I/O point range of the CPU module and network module (for remote I/O station).

(3) Multiple CPU system

When using the Ethernet module in a multiple CPU system, first refer to the QCPU User's Manual (Multiple CPU System).

(a) Applicable Ethernet module

When using the Ethernet module in a multiple CPU system, use the Ethernet module of function version B or later.

(b) Network parameter

Network parameters must be written only to the control PLC of the Ethernet module.

2 - 1

(4) Applicable software packages

(a) Software for PLC (*1)

The following table shows the systems and software packages applicable to the Ethernet module.

When using the Ethernet module, GX Developer is required.

		Software version
		GX Developer
Q00J/Q00/Q01CPU	Single CPU system	Version 7 or later
	Multiple CPU system	Version 8 or later
Q02/Q02H/Q06H/ Q12H/Q25HCPU	Single CPU system	Version 4 or later
	Multiple CPU system	Version 6 or later
Q12PH/Q25PHCPU	Single CPU system	Version 7.10L or later
	Multiple CPU system	
Q12PRH/Q25PRHCPU	Redundant system	Version 8.18U or later
When installing to MELSECNET/H remote I/O station		Version 6 or later

*1 Refer to Section 2.7 for the GX Developer versions that support the additional functions of the improved Ethernet module.

(b) Communication support tool for external devices

Item name	Model	Remark
MX Component	SWnD5C-ACT-E	Active X control library. The "n" in the model name is 0 or greater. (*1)

*1 Depending on the version of MX Component used, different versions of Ethernet modules are supported.
Refer to the manual of MX Component for the details.

2 - 2

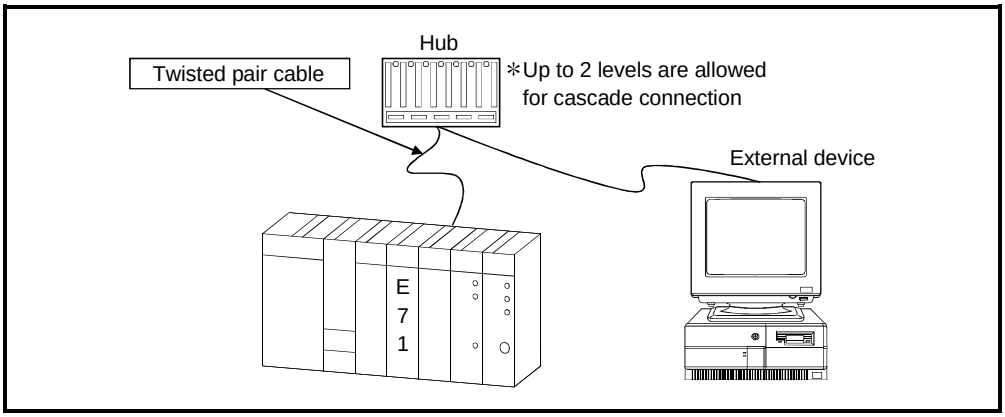
2.2 Devices Required for Network Configuration

This section explains the devices that are required to configure a network.
Network installation work requires sufficient safeguard; ask a network specialist for installation.

(1) When configuring an Ethernet system with a QJ71E71-100

When connecting a QJ71E71-100 to a network, either a 10BASE-T or 100BASE-TX can be used.
The Ethernet module detects whether it is 10BASE-T or 100BASE-TX, and the full-duplex or half-duplex transmission mode according to the hub.
For connection to the hub without the auto detection function, set the half-duplex mode on the hub side.

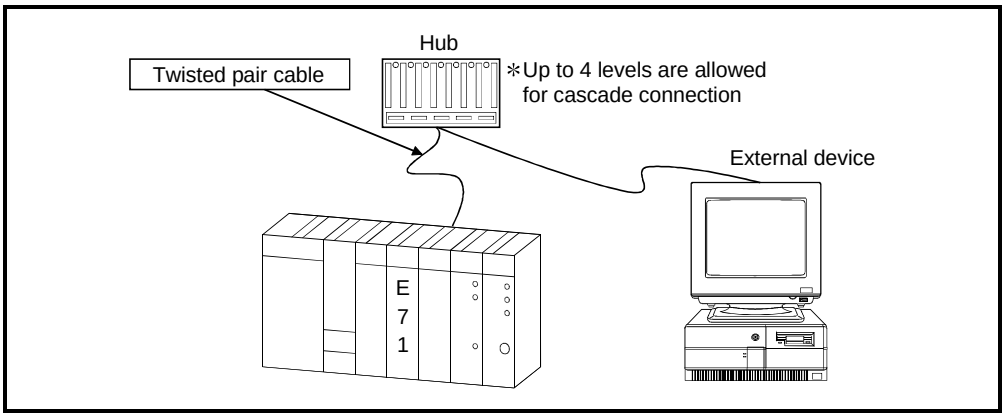
(a) Connection using the 100BASE-TX



Use devices that satisfy the standards of IEEE802.3 and 100BASE-TX.
(About the devices under the hub in the illustration above)

- Shielded twisted pair cable (STP cable), category 5
  * Straight cables can be used.
    (Correct operation is not guaranteed if a crossed cable is used to connect to an external device via the 100BASE-TX of the Ethernet module. However, it is possible to use crossed cables to connect two Ethernet modules (i.e., two QJ71E71-100 modules) for data communication or to connect an Ethernet module to a GOT.)
- RJ45 jacks
- 100Mbps hub

(b) Connection using the 10BASE-T



Use devices that satisfy the standards of IEEE802.3 and 10BASE-T.
(About the devices under the hub in the illustration above)

- Unshielded twisted pair cable (UTP cable), category 3 (4, 5)
  * Straight cables can be used.
    (Correct operation is not guaranteed if a crossed cable is used to connect to an external device via the 10BASE-T of the Ethernet module. However, it is possible to use crossed cables to connect two Ethernet modules (i.e., two QJ71E71-100 modules) for data communication or to connect an Ethernet module to a GOT.)
- RJ45 jacks
- 10Mbps hub

POINT

During the high-speed communication (100 M bps) via 100BASE-TX connection, a communication error may occur due to the effect of high frequency noise from devices other than PLC in a given installation environment.
The following describes countermeasures on the QJ71E71-100 side to prevent the effect of high frequency noise for construction of a network system.

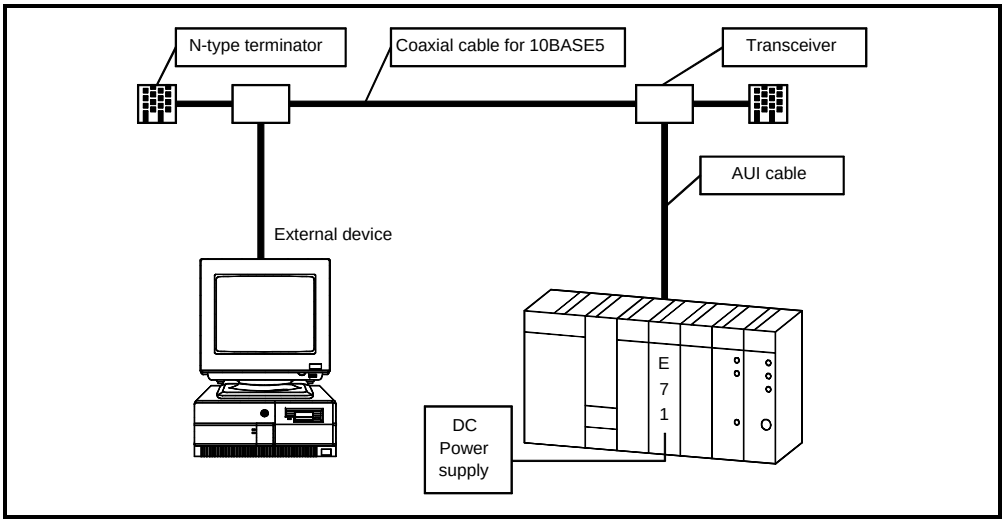
(1) Wiring connection
- Do not bundle the twisted pair cables with the main circuit and power wires, and do not install them close to each other.
- Make sure to place the twisted pair cables in a duct.

(2) Communication method
- Data communication with an external device is performed using TCP/IP communication.
- Increase the number of communication retries as necessary.

(3) 10 M bps communication
- Communication is performed at a data transmission rate of 10 M bps by changing the connection hub for the QJ71E71-100 to a hub capable of handling 10 M bps.

(2) When configuring the Ethernet system with a QJ71E71-B5

(a) Connecting using the 10BASE5



1) Use a 10BASE5 coaxial cable, an N-type terminator, a transceiver, and an AUI cable (transceiver cable) that meet the Ethernet standards.
2) Use a transceiver with SQETEST (Signal Quality Error TEST) or a heart beat function.
3) Use a DC power supply (power supply for transceiver) that meets the specifications of the transceiver and the AUI cable. (Refer to REMARK.)

REMARK

The transceiver power characteristics are as follows:
- Input terminal voltage: 12 $V^{-6\%}$ to 15 $V^{+5\%}$
- AUI cable direct resistance: 40 Ω/km or less, maximum length 50 m (164 ft.)
- Maximum current consumption: 500 mA or less

Thus, the applicable transceiver supply power will be from 13.28 V to 15.75 V.

✻ Calculation of the transceiver supply power's voltage drop (V)

Voltage drop (V) = AUI cable direct current resistance (Ω/m) × AUI cable length (m) × 2 (both directions) × transceiver consumption current (A)

(Example)

2.0 (V) = 0.04 (Ω/m) × 50 (m) × 2 × 0.5 (A)

In this case, the recommended transceiver supply power is more than 13.28 V.

13.28 (V) = 12 $V^{-6\%}$ (11.28 V) + 2.0 (V)

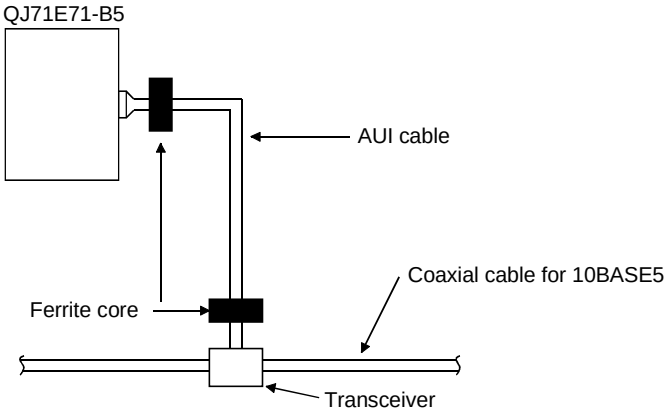
POINT

(1) Consult a network specialist for required devices.

(2) The following can be used as a countermeasure for errors due to high-frequency noise according to the installation environment.
- Mount a ferrite core using the method shown in (3) below.
- Increase the retry number when communicating by TCP/IP.

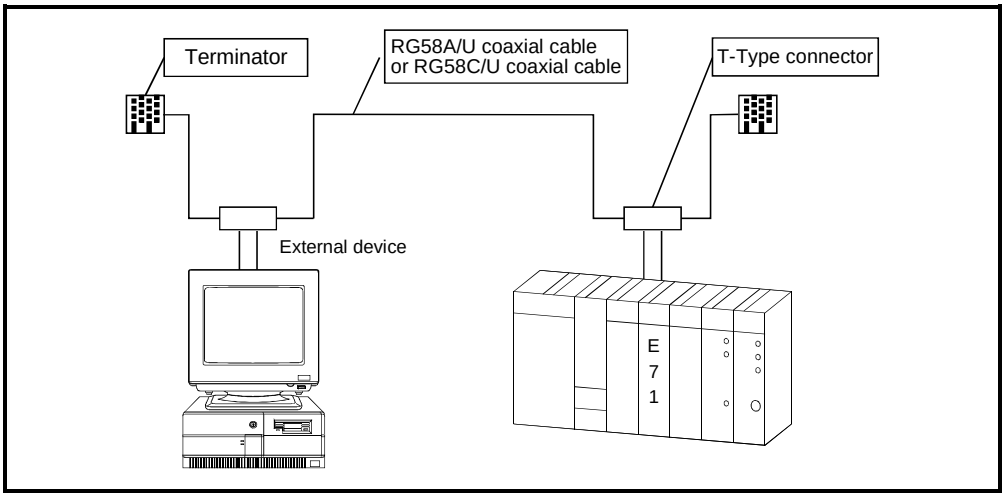
(3) Use the following method to mount a ferrite core when connected to the network by 10BASE5.

(Mounting the ferrite core)
Mount the ferrite core (*1) to the Ethernet module side and the external device side/the transceiver side for AUI cable.
*1 ZCAT 2032-0930 manufactured by TDK Corporation is usable.



(3) When configuring the Ethernet system with a QJ71E71-B2

(a) Connecting using the 10BASE2



Use devices that meet the standards of IEEE802.3 and 10BASE2.

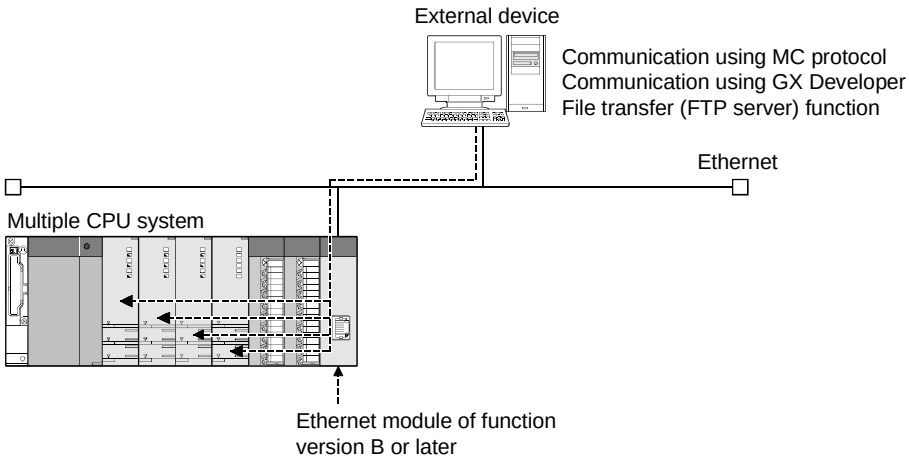
- RG58A/U or RG58C/U (coaxial cable 50 Ω)
- BNC-type Terminator (product equivalent to 221629-4 manufactured by Tyco Electronics AMP K. K.)
- T-shaped adapter (product equivalent to UG-274/U(1S) manufactured by Hirose Electric Co., Ltd.)

POINT
Consult a network specialist for required devices.

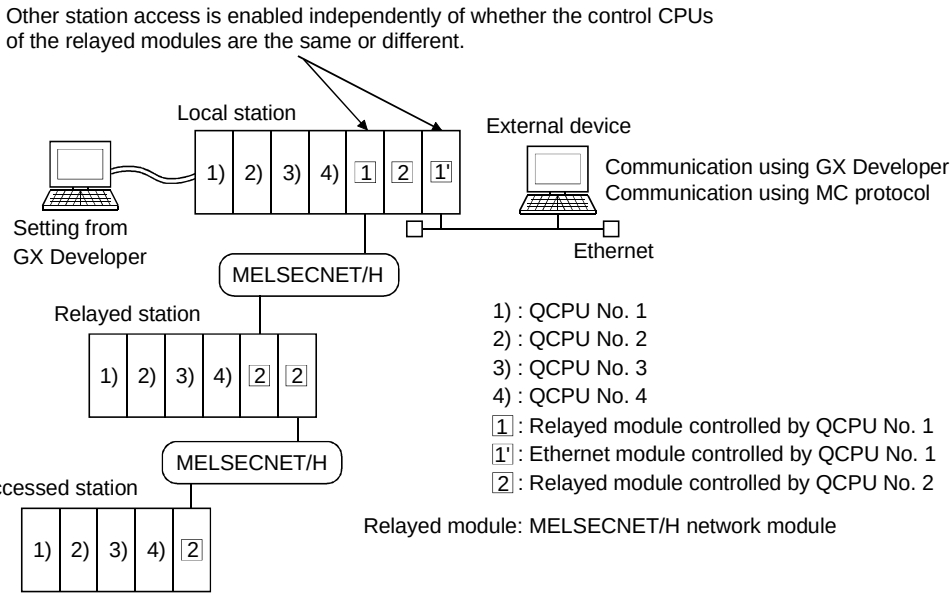
2.3 For Use in Multiple CPU System

This section describes the use of the Ethernet module in a multiple CPU system.

(1) When making access from the external device to the non-control CPU of the Ethernet module using any of the following functions, use the Ethernet module of function version B or later.
When the Ethernet module of function version A is used, only the control CPU can be accessed. (Access to the non-control CPU will result in an error.)
- Communication using MC protocol
- Communication using GX Developer
- File transfer (FTP server) function



(2) When the other station at the access destination is in a multiple CPU system and access is to be made to the non-control CPU of the relayed module of the accessed station, use the modules of function version B or later as the relayed modules and QCPUs of the local station and all the relayed and accessed stations. ∗1
(Example)


Other station access is enabled independently of whether the control CPUs of the relayed modules are the same or different.
Local station
External device
1) 2) 3) 4) 1 2 1'
Communication using GX Developer
Communication using MC protocol
Setting from GX Developer
Ethernet
MELSECNET/H
Relayed station
1) 2) 3) 4) 2 2
MELSECNET/H
Accessed station
1) 2) 3) 4) 2
1) : QCPU No. 1
2) : QCPU No. 2
3) : QCPU No. 3
4) : QCPU No. 4
1 : Relayed module controlled by QCPU No. 1
1' : Ethernet module controlled by QCPU No. 1
2 : Relayed module controlled by QCPU No. 2
Relayed module: MELSECNET/H network module

∗1 The targets of the relayed modules for other station access are as follows.
- MELSECNET/H, MELSECNET/10 network modules
- Q series C24
- Ethernet module

2.4 For Use with Q00J/Q00/Q01CPU

This section describes the use of the Ethernet module with the Q00J/Q00/Q01CPU.

(1) Available functions

The following functions are available when the Ethernet module is mounted on the main base unit of the Q00J/Q00/Q01CPU.

Function	Availability
Communication using the MC protocol	○(*1)
Communication using the fixed buffer	○
Reception processing by an interrupt program	○(*3)
Communication using the random access buffer	○
Sending/receiving of e-mail	○(*3)
Communication using the data link instruction	△(*2)
Reception processing by an interrupt program	○(*3)
File transfer (FTP server function)	○
Communication using the Web function	○
MELSECNET/H, MELSECNET/10 relay communication	○
Router relay communication (Router relay function)	○
External device existence confirmation	○
Pairing open communication	○
Communication via automatic open UDP port	○
Simultaneous broadcast	○
Support for the QCPU remote password function	○(*3)
Setting the Ethernet parameters through GX Developer	○
Access to QCPU with GX Developer (TCP/IP or UDP/IP)	○

○: Available △: Available with restriction ×: Unavailable

*1 For the number of its processing time, refer to the Reference Manual.
The devices that can be accessed and its range are different according to the frame used for data communication.

*2 When the target station of the SREAD/SWRITE instruction is the Q00J/Q00/Q01CPU, the read reporting device to the target station set to the argument (D3) is ignored.
The operation of the SREAD/SWRITE instruction is the same as that of the READ/WRITE instruction.
For the SREAD/SWRITE instruction, refer to Chapter 4 or Chapter 6 of the User's Manual (Application).

*3 Available for the Q00J/Q00/Q01CPU (function version B) or later.
To use the function, GX Developer Version 8 or later is required.

2.5 For Use with Q12PRH/Q25PRHCPU

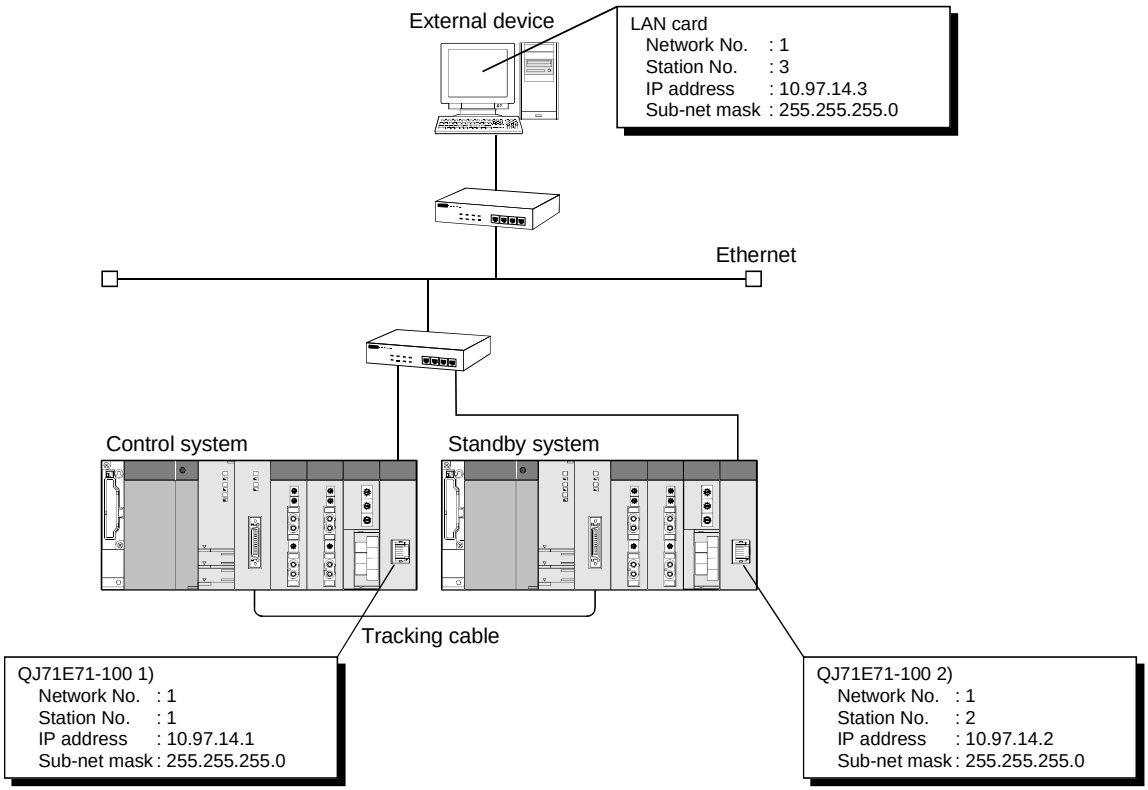
This section describes the use of the Ethernet module with the Q12PRH/Q25PRHCPU.

(1) System configuration

The configuration of a redundant system is shown below.

(a) Basic configuration of redundant system

The basic system configuration of the redundant system is shown below. Access can be made from the external device to the control system/standby system of the redundant system.
(Refer to Section 5.11 when configuring a network with the redundant system.)

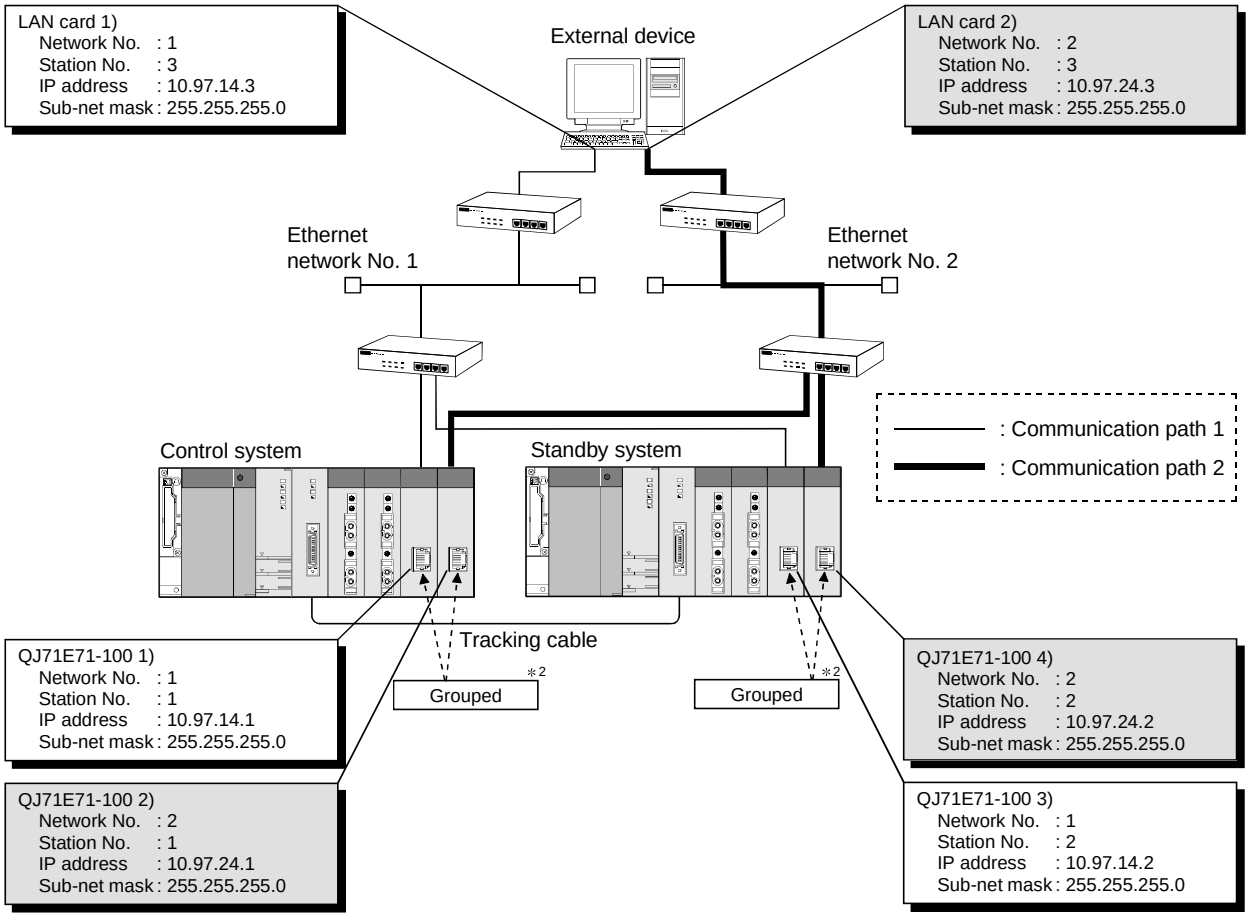


(b) System configuration with communication path backed up*1

The following shows a system configuration having a pair of communication paths (backed up) that connect the external device and Ethernet module. Access can be made from the external device to the control system/standby system of the redundant system in the communication path 1 or communication path 2.

(Example) If a communication error occurs during access to the control system in the communication path 1, access can be made to the control system in the communication path 2. *2

Further, if a communication error occurs in the communication path 2, system switching is performed between the control system and standby system, and communication can be continued to the new control system. *3



*1 When using two LAN cards in the external device, use them at different sub-net addresses to the set IP address using the sub-net mask.

*2 Perform "Network module redundant group settings" on GX Developer. (Refer to the QnPRHCPU User's Manual (Redundant System) for "Network module redundant group settings ".)
The system switches when both of the grouped Ethernet modules become faulty.

*3 Make the redundant setting of GX Developer to set whether a system switching request will be issued or not from the Ethernet module at detection of a communication error or disconnection. (Refer to Section 5.11.3.)

(2) Available functions

The following functions are available when the Ethernet module is mounted on the main base unit of the Q12PRH/Q25PRHCPU.

Function	Availability
Communication using the MC protocol	△
Communication using the fixed buffer	△
Communication using the random access buffer	△
Sending/receiving of e-mail	△
Communication using the data link instruction	△
File transfer (FTP server function)	△
Communication using the Web function	○
MELSECNET/H, MELSECNET/10 relay communication	△
Router relay communication (Router relay function)	○
External device existence confirmation	○
Pairing open communication	○
Communication via automatic open UDP port	○
Simultaneous broadcast	△
Support for the QCPU remote password function	○
Setting the Ethernet parameters through GX Developer	○
Access to QCPU with GX Developer (TCP/IP or UDP/IP)	○

○: Available △: Available with restriction (∗1) ×: Unavailable

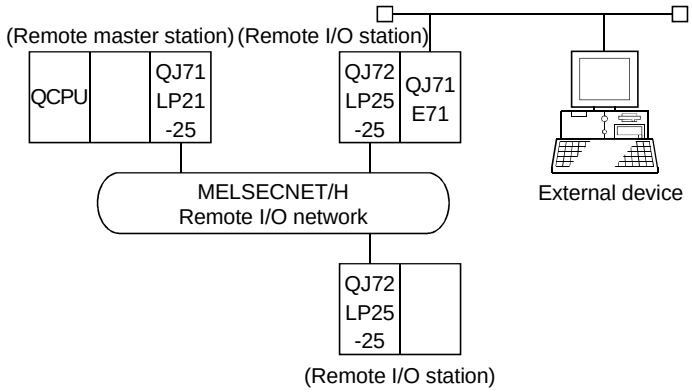
∗1 Refer to Section 5.11.5 for the precautions for using the functions on the Q12PRH/Q25PRHCPU.

2.6 For Use at MELSECNET/H Remote I/O Station

This section describes the use of the Ethernet module at a MELSECNET/H remote I/O station.
When using the Ethernet module with the QCPU, it is not necessary to read this section.

(1) System configuration

(Example)



(2) Available functions

The following functions are available when the Ethernet module is mounted on the MELSECNET/H remote I/O station.

Function		Availability
Initial processing	Sequence program	×
	GX Developer parameter setting	○ (See (3) (4))
Open/close processing	Sequence program	×
	GX Developer parameter setting	○ (See (3) (4))
Communication using the MC protocol		○ (*1)
Communication using the fixed buffer		×
Communication using the random access buffer		○
Sending/receiving of e-mail		×
Communication using data link instruction		× (Can be relayed)
File transfer (FTP server)		×
Communication using the web function		×
MELSECNET/H, MELSECNET/10 relay communication		○
Router relay communication (Router relay function)		○
External device existence confirmation		○
Pairing open communication		×
Communication while the automatic open UDP port		○
Simultaneous broadcast		×
Support for the QCPU remote password function		○ (*2)
Setting the Ethernet parameters through GX Developer		○ (See (3))
Access to QCPU with GX Developer (TCP/IP or UDP/IP)		○

○: Available ×: Unarailable

*1 The following explains access made to the MELSECNET/H remote I/O station using the MC protocol and other station access via the MELSECNET/H remote I/O station.

(a) Perform communication using a QnA compatible 3E frame. (Communication cannot be done with the A compatible 1E frame.)

(b) The following functions are available for the MELSECNET/H remote I/O station. For the QnA/A series compatible MELSECNET/10 remote I/O station, only read/write of the intelligent function module buffer memory can be performed.

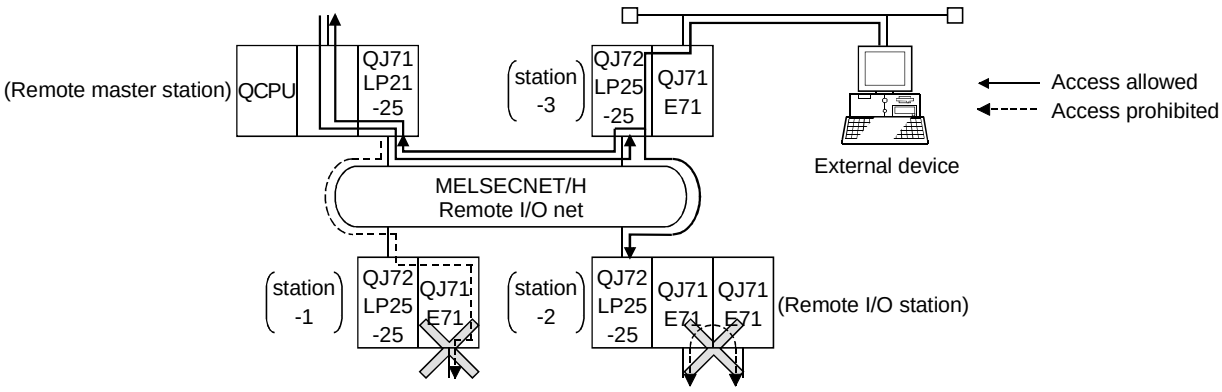
Available function	Function
Read/write of device memory	Batch read, batch write
	Random read, test (random write)
	Monitor data registration, monitor
	Multiple-block batch read, multiple-block batch write
Read/write of buffer memory	Read/write of Ethernet module buffer memory
Read/write of intelligent function module buffer memory	Read/write of specified intelligent function module buffer memory

(c) The following devices of the MELSECNET/H remote I/O station are accessible by read/write of the device memory.
See the Reference Manual for a detailed explanation.

Device name	Device symbol	Device name	Device symbol
Special relay	SM	Link relay	B
Special register	SD	Data register	D
Input relay	X	Link register	W
Output relay	Y	Link special relay	SB
Internal relay	M	Link special register	SW

(d) Other station access via the MELSECNET/H remote I/O station enables access to the MELSECNET/H remote master station and access to the MELSECNET/H remote I/O station.

1) In the following figure, access can be made from the external device to the remote master station and to the MELSECNET/H remote I/O station.



2) Other station access via the following MELSECNET/H remote I/O stations cannot be made.
- Other station access from the MELSECNET/H remote master station via the Ethernet module mounted on the MELSECNET/H remote I/O station. (Station 1 in the above figure)
- Other station access via between the Ethernet modules. (Station 2 in the above figure)

*2 This function is available for the MELSECNET/H remote I/O station of function version D and later.
GX Developer Version 8.18U or later is required to use the function.

(3) Setting parameters for the Ethernet module through GX Developer

The following parameters should be set through GX Developer in order to use the Ethernet module mounted on MELSECNET/H remote I/O station.
The setting can be made in the same way as when setting the parameters for the Ethernet module mounted on a QCPU station. See Section 4.5 and later.
See the operating manual for GX Developer for how to display each setting screen.
(Parameter setting items for the Ethernet module mounted on MELSECNET/H remote I/O station)

Parameter setting item		Setting	Remarks
I/O assignment		Set the module mounting information.	Setting not required
Network Parameters Setting the Number of MNET/10H Ethernet Cards		Perform settings to allow the Ethernet module to be used as a network module.	—
	Operational settings	Perform settings for items common to Ethernet modules.	
	Initial settings	Set timer values for data communication.	
	Open settings	Perform settings relating to open/close processing for connections.	
	Router relay parameter	Perform settings when communicating using routing relays.	
	Station No. <-> IP information	Perform settings when communicating via the MELSECNET/H, MELSECNET/10.	
Remote password settings		Perform settings for the remote password.	

POINT

(1) Connect GX Developer to MELSECNET/H remote I/O station and use it to set the parameters.

(2) Always set the "Network parameters setting the number of MNET/10H Ethernet cards" and "Operational settings."

(3) Reset MELSECNET/H remote I/O station after changing the setting.

REMARK

It is not necessary to set the “Intelligent function module switch settings” with GX Developer’s I/O assignment.
Each type of setting corresponding to the switch settings is performed in the above mentioned “Operational settings,” “Initial settings, “ and “Open settings.”

(4) Data communication procedure

(a) Connect GX Developer to MELSECNET/H remote I/O station and perform the following settings:

1) Communicating with TCP/IP
- Select "Always wait for OPEN" for the "Initial Timing" under the "Operational Settings."
- Select "Unpassive or Full passive" for the "Open system" setting under "Open settings."

2) Communicating with UDP/IP (when using the user port)
Select "Always wait for OPEN" for the "Initial Timing" under the "Operational Settings."
Data communication can also be done using the automatic open UDP port for the Ethernet module.

(b) Start up MELSECNET/H remote I/O station.

(c) Perform the open processing.

1) When communicating using TCP/IP, always perform "Active open" from the external device. (The status of the Ethernet module side is "Wait for open" because of "Passive open.")

2) When communicating using UDP/IP, perform the open processing on the external device side. (The status of the Ethernet module side is "Data communication enabled" through internal processing.)

(d) Perform data communication.

(e) After completing data communication, perform the close processing.

1) When communicating using TCP/IP, always perform the close processing from the external device. (The Ethernet module side performs the close processing at a request from the external device and its status becomes able to receive another open request.)

2) When communicating using UDP/IP, perform the close processing for the external device. (The close processing on the Ethernet module side is not required.)

2.7 Checking the Function Version and Serial No.

This section explains the function version, serial No. and how to check them for related products that can use functions added through improvements in the Ethernet module.

(1) Compatibility with related products designed for using functions added to the Ethernet module

Added function		Version of Ethernet module			Version of related product	
		QJ71E71-100	QJ71E71-B5	QJ71E71-B2	CPU module	GX Developer
Support for IEEE802.3 frames		○	○	Function version B or later, whose first 5 digits of the serial No. are 03102	○	Version 7 or later
Re-initial processing of the Ethernet module	Re-initial processing using sequence programs	Function version B or later, whose first 5 digits of the serial No. are 03102	○	Function version B or later, whose first 5 digits of the serial No. are 03061	○	○
	Re-initial processing via dedicated instruction (UINI instruction)			Function version B or later, whose first 5 digits of the serial No. are 03102		
	TCP Maximum Segment split transmission	Function version B or later, whose first 5 digits of the serial No. are 05051		Function version B or later, whose first 5 digits of the serial No. are 05051		
Re-open processing of the Ethernet module (∗ 1)		Function version B or later, whose first 5 digits of the serial No. are 05051	○	Function version B or later, whose first 5 digits of the serial No. are 05051	○	○
Existence check function	Checking by KeepAlive	Function version B or later, whose first 5 digits of the serial No. are 05051	○	Function version B or later, whose first 5 digits of the serial No. are 05051	○	Version 8.05F or later
Connection of up to 17 MELSOFT products via TCP/IP communication		○	○	Function version B or later, whose first 5 digits of the serial No. are 02122	○	Version 6.05F or later
Simplifying connection with MELSOFT products	Simplifying access to other stations	Function version B or later, whose first 5 digits of the serial No. are 05051	○	Function version B or later, whose first 5 digits of the serial No. are 05051	○	○
	Access with the same station number					
Ethernet diagnostic function of GX Developer	Monitoring of various Ethernet module statuses	○	○	Function version B or later	Function version A or later, whose first 5 digits of the serial No. are 02092	Version 6 or later
	PING test/loopback test via Ethernet board					
	PING test via CPU					Version 7 or later
Hub connection status monitor function		Function version D or later	×	×	○	○
When using the e-mail function	Sending files in CSV format as attachment	○	○	Function version B or later	○	Version 6 or later
	Sending main text			Function version B or later, whose first 5 digits of the serial No. are 03102		Version 7 or later
	Support for encoding/decoding	Function version B or later, whose first 5 digits of the serial No. are 03102		Function version B or later, whose first 5 digits of the serial No. are 03102		○
Target station CPU type specification for data link instructions		Function version D or later	Function version D or later	Function version D or later	○	○
Multiple CPU system compatibility of file transfer (FTP server) function		○	○	Function version B or later, whose first 5 digits of the serial No. are 03102	Function version B or later	○
Communication using the Web function		○	○	Function version B or later, whose first 5 digits of the serial No. are 05051	○	○
Remote password check		○	○	Function version B or later	Function version A or later, whose first 5 digits of the serial No. are 02092	Version 6 or later

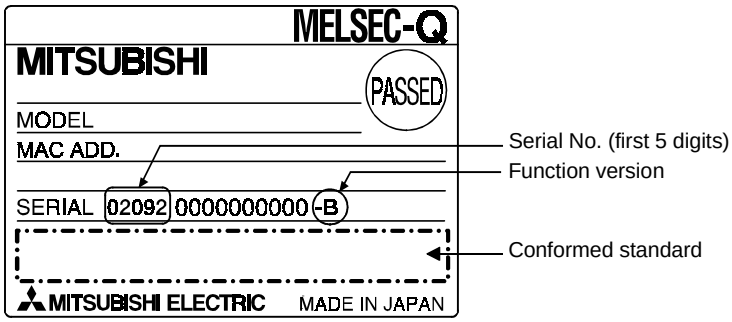
○ : Available (no restrictions on version) × : Unavailable

∗ 1 Change was made to the operation of the Ethernet module to be performed when the Active open request is received again from the external device in the open completion status of TCP. (Refer to POINT in Section 5.6 (2).)

(2) Checking the function version and serial No. of the Q series PLC

(a) Checking using the "Rating plate" on the side of the module

The serial No. and function version of the subject module are displayed in the SERIAL line of the rating plate.



(b) Checking through GX Developer

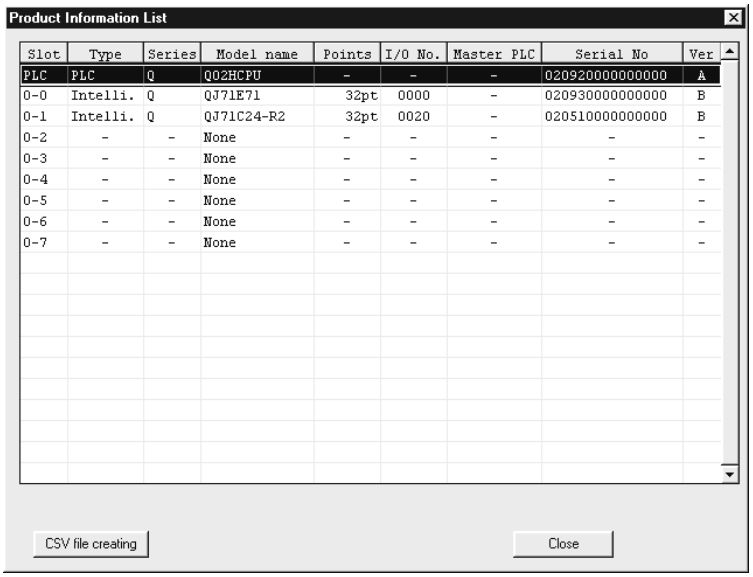
The following explains how to check the serial No. and function version of the target module through GX Developer.

The serial No. and function version are displayed on the "Product Information List" or the "Module's Detailed Information" screen of GX Developer.

The following diagram shows how to check the serial No. and function version on the "Product Information List" screen. (See Section 11.2 for the "Module's Detailed Information" screen.)

[Startup procedure]

"Diagnostics" → "System monitor" → "Product Inf. List"


Product Information List
Slot | Type | Series | Model name | Points | I/O No. | Master PLC | Serial No | Ver
PLC | PLC | Q | Q02HCPU | - | - | - | 0209200000000000 | A
0-0 | Intelli. | Q | QJ71E71 | 32pt | 0000 | - | 0209300000000000 | B
0-1 | Intelli. | Q | QJ71C24-R2 | 32pt | 0020 | - | 0205100000000000 | B
0-2 | - | - | None | - | - | - | - | -
0-3 | - | - | None | - | - | - | - | -
0-4 | - | - | None | - | - | - | - | -
0-5 | - | - | None | - | - | - | - | -
0-6 | - | - | None | - | - | - | - | -
0-7 | - | - | None | - | - | - | - | -
CSV file creating
Close

[Serial No., Ver.]

- The serial No. of the target module is displayed in the serial No. area.
- The function version of the target module is displayed in the Ver. area.

3 SPECIFICATIONS

This section explains the Ethernet module performance specifications and transmission specifications.
For more details on the general specifications, refer to the User's Manual for the QCPU (Q mode).

3.1 Performance Specifications

The following explains the performance specifications of the Ethernet module.

Item			Specification			
			QJ71E71-100		QJ71E71-B5	QJ71E71-B2
			100BASE-TX	10BASE-T	10BASE5	10BASE2
Transmission specifications	Data transmission speed		100 M bps	10 M bps		
	Communication mode		Full-duplex/Half-duplex		Half-duplex	
	Transmission method		Base band			
	Maximum node-to-node distance		—		2500 m (8202.10 ft.)	925 m (3034.77 ft.)
	Maximum segment length		100 m (328.08 ft.) (*1)		500 m (1640.42 ft.)	185 m (606.96 ft.)
	Maximum number of nodes/connection		Cascade connection Maximum 2 stages	Cascade connection Maximum 4 stages	100 units/segment	30 units/segment
	Interval between the minimum nodes		—		2.5 m (8.20 ft.)	0.5 m (1.64 ft.)
Transmission data storage memory	Number of simultaneously open connections allowed		16 connections (Connections usable by the sequence program)			
	Fixed buffer		1 k words × 16			
	Random access buffer		6 k words × 1			
	E-mail	Attached file	6 k words × 1 (*4)			
		Main text	960 words × 1 (*4)			
Number of I/O points occupied			32 points/1 slot (I/O assignment: intelligent)			
5 V DC internal current consumption			0.50 A		0.50 A	0.60 A (*3)
12 V DC external power supply capacity (Transceiver)			—		(*2)	—
External dimensions			98 (3.86 in.) (H) × 27.4 (1.08 in.) (W) × 90 (3.54 in.) (D) [mm]			
Weight			0.11 kg (0.24 lb)		0.12 kg (0.26 lb)	0.13 kg (0.29 lb) (*3)

*1 Length between the Hub and nobe.

*2 It is necessary to apply a transceiver, or a device that meets AUI cable specifications. (Refer to Section 2.2)

*3 The product with first 5 digits of serial number "05049" or earlier is different as follows:
- 5V DC internal current consumption: 0.70A
- Weight: 0.14kg (0.31lb.)

*4 The following table outlines the specifications of the e-mail transmission and reception function.
The e-mail transmission and reception function of the PLC CPU and the random access buffer communication function cannot be used together; only one of the functions can be used at a time. (The e-mail transmitting function of the Ethernet module's PLC CPU monitoring function and the random access buffer communication function can be used together.)

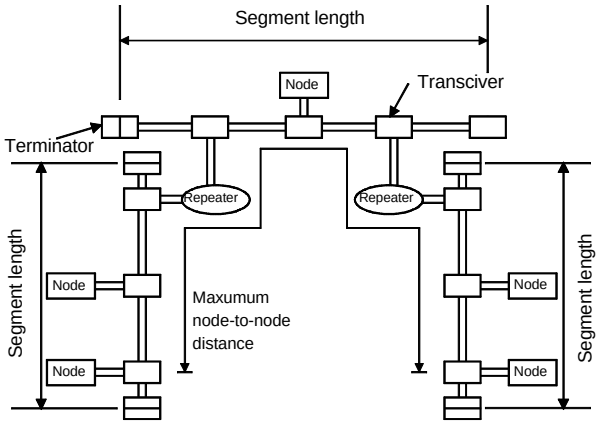
Item			Specification
Transmission specifications Transmission and reception data	Data size	Attached file	6 k words x 1
		Main text	960 words x 1
	Data transfer method		When sending: Sends either a file as attachment or main text (select one). When receiving: Received a file as attachment.
	Subject		Us-ASCII format or ISO-2022-JP (Base64)
	Attached file format		MIME format
	MIME		Version 1.0
	Data of attached file format		Binary/ASCII/CSV can be selected. File name: XXXX.bin (binary), XXXX.asc (ASCII), XXXX.csv (CSV) (CSV: Comma Separated Value)
	Division of attached file		Cannot be divided (only one file can be sent/received) * If any divided files are received, only the first file will be received and the remaining files will be discarded.
	When sending (encode)		Subject : Base64/7 bits Main text : 7 bits Attached file : Base64
	When receiving (decode)		Subject : (Does not decode) Main text : (Cannot be received) Attached file : Base64/7 bits/8 bits/Quoted Printable * If e-mail is sent from the external device to the PLC side, specify the encoding method (Base64/7 bits/ 8 bits/Quoted Printable) of the attached file.
	Encryption		No
	Compression		No
	Communication with mail server		SMTP (sending server) Port number = 25 POP3 (receiving server) Port number = 110
	Operation check mailer		Microsoft® Corporation Internet Explorer 5.0 (Outlook Express 5.5/Outlook Express 5) Netscape® Communications Corporation Netscape® 4.05

3

REMARK

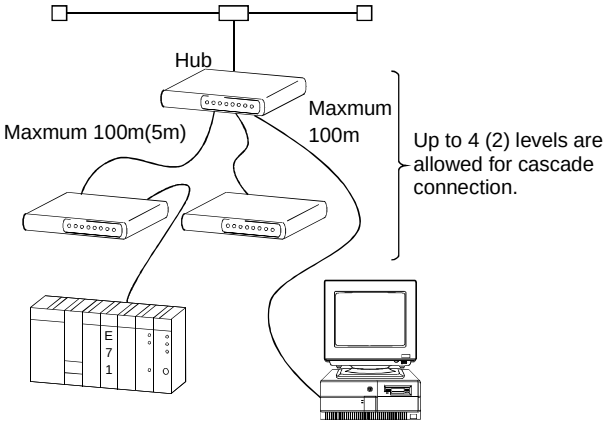
The following explains each of the transmission specification items.

[Connecting using the 10BASE2/10BASE5]



* There is no transceiver when connecting using the 10BASE2.

[Connecting using the 10BASE-T/100BASE-TX]



The item in parentheses ( ) indicates when a connection is established using a 100BASE-TX.

3.2 Data Codes for Communication

This section explains the data codes used in the communication between the Ethernet module and the external device or the PLC CPU.

(1) The data codes used while communicating are listed below.

1) Ethernet module ↔ External device
Data can be communicated by selecting either binary code or ASCII code in the data code setting of GX Developer, as shown below.
The switching between binary code and ASCII code is set using GX Developer:
[GX Developer] - [Network parameter] - [Operational settings] - [Communication data code]
For more details, see the Section 4.7 "Operational Settings."

Data communication function		Communication data code		Reference chapter
		Binary code	ASCII code	
Communication using MC protocol	Automatic open UDP port	○ (＊1)	—	Chapter 6
	User open port	○	○	
Communication using fixed buffer	Procedure exist	○	○	Chapter 7
	No procedure	○ (＊1)	—	Chapter 8
Communication using random access buffer		○	○	Chapter 9

○: Selectable ×: Cannot be communicated

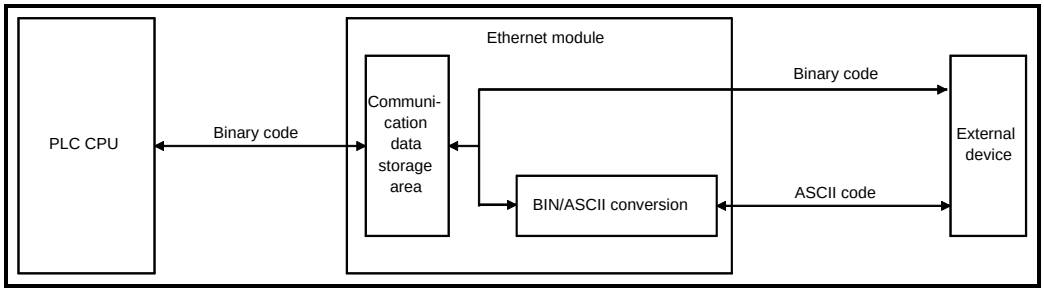
＊1 Communication is performed in binary code regardless of the setting in [Operational Settings] – [Communication data code] (see Section 4.7).

REMARK

When the following data communication functions are used, communication is performed using the data code handled by each function, regardless of the setting of the communication data code.

- Sending/receiving e-mail
- MELSECNET/H, MELSECNET/10 relay communication
- Communication using data link instructions
- File transfer (FTP server function)
- Communication using the Web function

2) Ethernet module ↔ PLC CPU
Data is sent and received in binary code.



(2) When communicating using ASCII code, 1-byte binary code data is automatically converted into 2-byte ASCII code data and then transmitted.

(Example)

Binary code data		ASCII code data
15_H (One byte)	→	31_H, 35_H "1", "5" (Two bytes)
1234_H (Two bytes)	→	31_H, 32_H, 33_H, 34_H "1", "2", "3", "4" (Four bytes)

(3) The size of data that can be communicated between the Ethernet module and an external device at a time is determined by the function used and the data code (binary/ASCII) that is set by selecting [GX Developer] – [Communication data code].
The following shows the maximum sizes of communication data that can be sent and received at a time with each data communication function.

Data communication function		Exchangeable data size
Communication using MC protocol		The maximum number of point that can be designated with each command/instruction : Maximum of 1920 bytes
Communication using fixed buffer	Procedure exist	1017 words (Binary code) 508 words (ASCII code)
	No procedure	2046 words
Communication using random access buffer		1017 words (Binary code) 508 words (ASCII code)
Sending/receiving by e-mail		Attached file : Maximum of 6 k words, Main text : Maximum of 960 bytes
MELSECNET/H, MELSECNET/10 relay communication		480 words
Communication using data link instruction		The maximum number of point that can be designated with each command/instruction : Maximum of 960 bytes
File transfer (FTP server function)		Maximum of one file
Communication using web function		The maximum number of point that can be designated with each command/instruction : Maximum 1920 bytes.

3.3 Relationship between the External Devices and Additional Functions for Each Communication Function

This section explains with which external devices data communication can be performed and which additional functions can be used for each function.

(1) Communicability with external devices using various functions

The following table lists the communicability with external devices using various functions.

Function	External device				
	Personal computer ↓ QJ71E71	Personal computer ↑ QJ71E71	QJ71E71 ↓ ↑ QJ71E71	QJ71E71 ↓ Conventional model	Conventional model ↓ QJ71E71
Communication using the MC protocol	○	×			
Communication using the fixed buffer	○				
Communication using the random access buffer	○	×			
Sending/receiving e-mail	○			×	
Communication using data link instructions	×		○	△ (＊1)	×
File transfer (FTP server function)	○	×			
Communication using the Web function	○	×			

○: Can communicate △: Can communicate (with restrictions) ×: Cannot communicate

QJ71E71: Q Series Ethernet interface module

Conventional model: QnA or A Series Ethernet interface module

＊1 Communication can be performed with the QnA Series Ethernet interface module.

(2) Relationship with additional functions

The following table lists the correspondence between functions and their additional functions that can be used.

Communication function		Additional function							Communication method	
		MELSECNET/H, MELSECNET/10 relay communication	Router relay communication (router relay function)	Existence check of external device	Communication via pairing open	Communication via automatic open UDP port	Remote password check	Simultaneous broadcast	TCP/IP	UDP/IP
Communication using the MC protocol	QnA compatible 3E frame	○	○	○ (＊2)	×	○	○	○ (＊1)	○	○
	A compatible 1E frame	×	○	○	×	×	○	×	○	○
Communication using the fixed buffer	Procedure exist	×	○	○	○	×	○	×	○	○
	No procedure	×	○	○	○	×	○	○ (＊1)	○	○
Communication using the random access buffer		×	○	○	×	×	○	×	○	○
Sending/receiving e-mail		×	×	×	×	×	×	×	○	×
Communication using data link instructions		○	○	×	×	○	×	○	×	○
File transfer (FTP server function)		×	○	×	×	×	○	×	○	×
Communication using the Web function		○	○	×	×	×	○	×	○	×

○: Available ×: Not available or this function does not correspond to any of the functions in the function column.

＊1 Valid only for UDP.

＊2 The automatic open UDP port is excluded.

3.4 Ethernet Module Function List

This section shows a list of Ethernet module functions.

(1) Basic functions of the Ethernet module

The Ethernet module can perform the communications shown in the table below via the TCP/IP or UDP/IP communication.

Function		Description	Reference section
Communication using the MC protocol	QnA compatible 3E frame	Reads/writes PLC CPU data from/to an external device.	Chapter 5 Reference Manual
	A compatible 1E frame		
Communication using the fixed buffer	Procedure exist	Sends/receives arbitrary data between the PLC CPU and the external device using the fixed buffer of the Ethernet module.	Chapter 7
	No procedure		Chapter 8
Communication using the random access buffer		Reads/writes data from multiple external devices to the random access buffer of the Ethernet module.	Chapter 9
Sending/receiving e-mail		Sends/receives data via e-mail. • Sending/receiving by the PLC CPU • Sending by the PLC CPU monitoring function (automatic notification function)	Chapter 2 of User's Manual (Application)
Communication using data link instructions		Reads/writes the PLC CPU data of other station via Ethernet using data link instructions.	Chapter 4 of User's Manual (Application)
File transfer (FTP server function)		Reads/writes in file units using the FTP command from the external device.	Chapter 5 of User's Manual (Application)
Communication using the Web function		Reads/writes the PLC CPU data via the Internet using a commercially available Web browser.	User's Manual (Web Function)

(2) Additional functions of the Ethernet module

The following table lists the additional functions of the Ethernet module that can be used.

Function	Description	Reference section
MELSECNET/H, MELSECNET/10 relay communication	In a network system on which an Ethernet and a MELSECNET/H or MELSECNET/10 coexist or in a network system that relays multiple Ethernets, data communication is performed via multiple number of such networks.	Chapter 3 of User's Manual (Application)
Router relay communication (router relay function)	Performs data communication via a router or gateway. (The router relay function is not a function by which the Ethernet module works as a router.)	Section 5.3
Existence check of external device (Existence check function)	Checks whether or not the external device is working normally after a connection is established (open processing).	Sections 5.2.2 and 5.5
Communication via pairing open	Pairs and then opens a reception connection and a transmission connection (for fixed buffer).	Section 5.7
Communication via automatic open UDP port	Enables communication after the station in which an Ethernet module is mounted has been started. (Open/close processing by sequence programs is not required.)	Section 5.8
Remote password check	Prevents unauthorized access by a remote user to a QCPU.	Section 5.9
Simultaneous broadcast	Sends/receives data for all external devices on the same Ethernet as for the Ethernet module in data communication via UDP/IP.	Section 8.5
Connecting MELSOFT products (GX Developer, MX Component etc.) and GOTs	MELSOFT products (GX Developer, MX Component etc.) and GOTs are connected via TCP/IP communication or UDP/IP communication. It is also possible to connect multiple MELSOFT products and GOTs simultaneously.	Section 1.2 (6) Manual of each MELSOFT product

(3) Status check of the Ethernet module

Whether the Ethernet module is working normally and whether it can communicate normally are checked.

Function	Description	Reference section
Self refrain test	Checks the Ethernet module's sending/receiving function and line connection status.	Section 4.8.1
Hardware test	Tests the RAM and ROM of the Ethernet module.	Section 4.8.2
Communication error storage	When a data communication error occurs, this function stores the error information (error log), including the message subheader and IP address of the external device for a maximum of 16 pairs in the buffer memory.	Chapter 11

3.5 Dedicated Instruction List

The following table lists the dedicated instructions that can be used by the Ethernet module.

Application		Instruction name	Description	Reference section
For opening and closing a connections		OPEN	Opens a connection.	Section 10.8
		CLOSE	Closes a connection.	Section 10.5
For reinitialization		UINI	Reinitializes the Ethernet module.	Section 10.9
For fixed buffer communication		BUFRCV	Reads data received in the fixed buffer.	Section 10.2
		BUFRCVS	Reads data received in the fixed buffer using an interrupt program.	Section 10.3
		BUFSND	Sends data to the external device using the fixed buffer.	Section 10.4
For reading and clearing error information		ERRCLR	Clears the error (turns off the [COM. ERR] LED, clears the error log).	Section 10.6
		ERRRD	Reads error information.	Section 10.7
For sending and receiving e-mail		MRECV	Receives e-mail.	Section 6.2 of User's Manual (Application)
		MSEND	Sends e-mail.	Section 6.3 of User's Manual (Application)
For other station PLC communication (instructions for data link)	For reading/writing device data	READ	Reads word devices of other stations.	Section 6.4 of User's Manual (Application)
		SREAD	Reads word devices of other stations (with complete device).	Section 6.10 of User's Manual (Application)
		WRITE	Writes to word devices of other stations.	Section 6.12 of User's Manual (Application)
		SWRITE	Writes to word devices of other stations (with complete device).	Section 6.11 of User's Manual (Application)
		ZNRD	Reads word devices of other stations.	Section 6.13 of User's Manual (Application)
		ZNWR	Writes word devices of other stations.	Section 6.14 of User's Manual (Application)
	For sending/ receiving messages (arbitrary data)	SEND	Sends data to other stations.	Section 6.9 of User's Manual (Application)
		RECV	Reads data received from other stations (for main program).	Section 6.5 of User's Manual (Application)
		RECVS	Reads data received from other stations (for interrupt programs).	Section 6.6 of User's Manual (Application)
	Reading/ writing clock data, remote RUN/STOP	REQ	Runs/stops other stations in remote mode.	Section 6.7 of User's Manual (Application)
			Reads/writes the clock data of other stations.	Section 6.8 of User's Manual (Application)

3.6 List of GX Developer Setting Items for Ethernet Modules

The following table lists the parameter setting items that are set using GX Developer.

Parameter setting item	Description of setting	Function and parameter setting requirement (*1)								Reference section (*2)
		MC	Fixed	Random	Data link	FTP	Automatic notification	Mail	Web	
PLC parameter	—	—								—
I/O assignment settings	Set the module mount information. This is set when an Ethernet module is used in a multiple CPU system.	△	△	△	△	△	△	△	△	Section 4.5
Interrupt pointer settings	Set the relationship between the control number (SI) on the Ethernet module side and the interrupt pointer used on the PLC CPU side.	—	△	—	△	—	—	—	—	Section 7.3
Network parameters Setting the number of MNET/10H Ethernet cards	Perform settings for using the Ethernet module as a network module.	○	○	○	○	○	○	○	○	Section 4.6
Operational settings	Set the common items of the modules. These settings are required for initial processing.	○	○	○	○	○	○	○	○	Section 4.7
Initial settings	Set the data communication timer values.	△	△	△	△	△	△	△	△	Section 5.2
	Set the DNS server's IP address.	—	—	—	—	—	△	△	—	Chapter 2 of Application
Open settings	Set up the open processing for connection in order to perform data communication with the external device.	○	○	○	—	—	—	—	—	Section 5.5
Router relay parameter	Set the router relay of Ethernet.	△	△	△	△	△	△	△	△	Section 5.3
Station No. <-> IP information	Perform settings so that Ethernet can be treated equivalent to a MELSECNET/H or MELSECNET/10 network system in order to communicate with PLCs of other stations.	△	—	—	△	—	—	—	△	Chapter 3 of Application
FTP parameters	Perform settings for file transfer (FTP).	—	—	—	—	○	—	—	—	Chapter 5 of Application
E-mail settings	Perform settings for sending/receiving e-mail and for using the automatic notification function.	—	—	—	—	—	○	○	—	Chapter 2 of Application
Send mail address setting	Set the destination mail address.	—	—	—	—	—	○	○	—	
News setting	Set the notification conditions.	—	—	—	—	—	○	—	—	
Interrupt settings	Set the control number (SI) on the Ethernet module side when requesting an interrupt to the PLC CPU.	—	△	—	△	—	—	—	—	Section 7.3
Redundant setting	Perform settings for using the Ethernet module in a redundant system.	—	—	—	—	—	—	—	—	Section 5.11.3
Group settings										Section 4.6
Routing parameters	Perform settings of the relay station so that Ethernet can be treated equivalent to a MELSECNET/H or MELSECNET/10 network system in order to communicate with PLCs of other stations.	△	—	—	△	—	—	—	△	Chapter 3 of Application
Remote password settings	Set the target connection of the remote password check.	△	△	△	△	△	—	—	△	Section 5.9.5

○: Must be set when the applicable function is used. △: Set as needed ×: Setting is not required.

*1 The meanings of the abbreviations used in the table above are as follows:
MC: Communication using the MC protocol
Random: Communication using the random access buffer
FTP: File transfer protocol
Mail: E-mail
Fixed: Communication using the fixed buffer
Data link: Communication using data link instructions
Automatic notification: Data transmission via automatic notification
Web: Communication using the Web function

*2 "Application" in the Reference Section column refers to the User's Manual (Application).

3.7 List of Input/Output Signals to/from the PLC CPU

This section explains the input/output signals of the Ethernet module.
For assignments of the input/output signals shown in the table below, it is assumed that the Ethernet module is mounted in slot 0 of the basic base unit.
Device numbers starting with X indicate input signals from the Ethernet module to the PLC CPU, and device numbers starting with Y indicate output signals from the PLC CPU to the Ethernet module.
The following shows the input/output signals to/from the PLC CPU.

Signal direction: Ethernet module → PLC CPU			Signal direction: PLC CPU → Ethernet module		
Device number	Signal name	Reference section	Device number	Signal name	Reference section
X0	For fixed buffer communication of connection No. 1 ON : Sending normal completion or reception completion OFF: —	—	Y0	Connection No. 1 ON : At sending request or reception complete confirmation signal OFF: —	—
X1	For fixed buffer communication of connection No. 1 ON: Detection of sending error or reception error OFF: —	—	Y1	Connection No. 2 ON : At sending request or reception complete confirmation signal OFF: —	—
X2	For fixed buffer communication of connection No. 2 ON : Sending normal completion or reception completion OFF: —	—	Y2	Connection No. 3 ON : At sending request or reception complete confirmation signal OFF: —	—
X3	For fixed buffer communication of connection No. 2 ON: Detection of sending error or reception error OFF: —	—	Y3	Connection No. 4 ON : At sending request or reception complete confirmation signal OFF: —	—
X4	For fixed buffer communication of connection No. 3 ON : Sending normal completion or reception completion OFF: —	—	Y4	Connection No. 5 ON : At sending request or reception complete confirmation signal OFF: —	—
X5	For fixed buffer communication of connection No. 3 ON: Detection of sending error or reception error OFF: —	—	Y5	Connection No. 6 ON : At sending request or reception complete confirmation signal OFF: —	—
X6	For fixed buffer communication of connection No. 4 ON : Sending normal completion or reception completion OFF: —	—	Y6	Connection No. 7 ON : At sending request or reception complete confirmation signal OFF: —	—
X7	For fixed buffer communication of connection No. 4 ON: Detection of sending error or reception error OFF: —	—	Y7	Connection No. 8 ON : At sending request or reception complete confirmation signal OFF: —	—
X8	For fixed buffer communication of connection No. 5 ON : Sending normal completion or reception completion OFF: —	—	Y8	Connection No. 1 ON : Open request OFF: —	—
X9	For fixed buffer communication of connection No. 5 ON: Detection of sending error or reception error OFF: —	—	Y9	Connection No. 2 ON : Open request OFF: —	—
XA	For fixed buffer communication of connection No. 6 ON : Sending normal completion or reception completion OFF: —	—	YA	Connection No. 3 ON : Open request OFF: —	—
XB	For fixed buffer communication of connection No. 6 ON: Detection of sending error or reception error OFF: —	—	YB	Connection No. 4 ON : Open request OFF: —	—
XC	For fixed buffer communication of connection No. 7 ON : Sending normal completion or reception completion OFF: —	—	YC	Connection No. 5 ON : Open request OFF: —	—
XD	For fixed buffer communication of connection No. 7 ON: Detection of sending error or reception error OFF: —	—	YD	Connection No. 6 ON : Open request OFF: —	—
XE	For fixed buffer communication of connection No. 8 ON : Sending normal completion or reception completion OFF: —	—	YE	Connection No. 7 ON : Open request OFF: —	—
XF	For fixed buffer communication of connection No. 8 ON: Detection of sending error or reception error OFF: —	—	YF	Connection No. 8 ON : Open request OFF: —	—

Signal direction: Ethernet module → PLC CPU			Signal direction: PLC CPU → Ethernet module		
Device number	Signal name	Reference section	Device number	Signal name	Reference section
X10	Open completed for connection No. 1 ON : Open complete signal OFF: —	—	Y10	Use prohibited	—
X11	Open completed for connection No. 2 ON : Open complete signal OFF: —	—	Y11		
X12	Open completed for connection No. 3 ON : Open complete signal OFF: —	—	Y12		
X13	Open completed for connection No. 4 ON : Open complete signal OFF: —	—	Y13		
X14	Open completed for connection No. 5 ON : Open complete signal OFF: —	—	Y14		
X15	Open completed for connection No. 6 ON : Open complete signal OFF: —	—	Y15		
X16	Open completed for connection No. 7 ON : Open complete signal OFF: —	—	Y16		
X17	Open completed for connection No. 8 ON : Open complete signal OFF: —	—	Y17	COM.ERR LED Off request ON : At off request OFF: —	Section 11.1
X18	Open abnormal detection signal ON : Abnormal detection OFF: —	Section 5.6	Y18	Use prohibited	—
X19	Initial normal completion signal ON : Normal completion OFF: —	Section 5.1	Y19	Initial request signal ON : At request OFF: —	—
X1A	Initial abnormal completion signal ON : Abnormal completion OFF: —	Section 5.1	Y1A	Use prohibited	—
X1B	Use prohibited	—	Y1B		
X1C	COM.ERR LED lit confirmation ON : Lit (See Section 11.1) OFF: Off	Section 11.1	Y1C		
X1D	Use prohibited	—	Y1D		
X1E			Y1E		
X1F	Watchdog timer error detection ON : Watchdog timer error OFF: —	Section 11.1	Y1F		

Important

Among the input/output signals for the PLC CPU, do not output (ON) the signals marked with "Use prohibited." If any of the "Use prohibited" signals is output, the PLC system may malfunction.

POINT

(1) The input/output signals shown in this section are used when the QnA series Ethernet interface module programs are also used for the Q series Ethernet module (see Section 2 in Appendix).
In the QCPU, the input/output signals for intelligent function modules are turned on/off with dedicated instructions.
It is not necessary to turn on/off the signals by a sequence program, except for those that are shown in the programming examples in each of function explanation sections.

(2) When the QnA series Ethernet interface module programs are also used for the Q series Ethernet module, it is also recommended to replace the instructions with the dedicated instructions shown in the corresponding function explanation section of each manual for the Q series Ethernet module.

3.8 List of Applications and Assignments of the Buffer Memory

This section explains details of the buffer memory.

(1) Configuration of the buffer memory

Buffer memory consists of a user area and a system area, as listed below.

(a) User areas

1) The areas where the user writes/reads data.

2) A user area consists of a parameter area for initial processing and data communication, an area for data communication, and an area for storing communication status and communication error data.

3) Reading/writing data to the user area should be performed according to the instructions in the corresponding detailed explanation section. Data communication may take long if continually executed; therefore, execute only when needed.

(b) System areas

The areas used by the Ethernet module

Important

Do not write data in the "system areas" of the buffer memory.
If data is written to any of the system areas, the PLC system may not be operated properly.
Some of the user areas contain partially system areas. Care must be taken when reading/writing to the buffer memory.

POINT

(1) Use the FROM/TO instructions or other applicable commands to access the buffer memory addresses shown in this section when the QnA series Ethernet module programs are used for the Q series Ethernet module (see Section 2 in Appendix).
In the QCPU, the buffer memory of an intelligent function module is accessed by dedicated instructions.
It is not necessary to access directly using the FROM/TO or other instructions from the sequence program, except when accessing the buffer memory as shown in the programming examples in each of function explanation sections.

(2) When the QnA series Ethernet module programs are used for the Q series Ethernet module, it is also recommended to replace the instructions with the dedicated instructions shown in the corresponding function explanation section of each manual for the Q series Ethernet module.

(2) Assignments of the buffer memory

A buffer memory consists of 16 bits per address.

<Bit configuration diagram>

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

The following shows the buffer memory addresses.

Address Decimal (Hexadecimal)	Application	Name		Initial value (Hexadecimal)	GX Developer setting applicability (*1)	Reference section (*2)
0 to 1 (0 to 1H)	Initial processing parameter setting area	Local station Ethernet module IP address		C00001FEH	○	Section 4.7
2 to 3 (2 to 3H)		System area		–	–	–
4 (4H)		Special function settings • Router relay function (b5, b4) 00: Do not use (default) 01: Use • Conversion system setting for MELSECNET/H, MELSECNET/10 relay function (b7, b6) 00: Automatic response system (default) 01: IP address computation system 10: Table exchange system 11: Use-together system • FTP function setting (b9, b8) 00: Do not use 01: Use (default) Bits other than above are reserved for system use.		0100H	○	Section 5.3 Chapter 3 of Application Chapter 5 of Application
5 to 10 (5 to AH)		System area		–	–	–
11 (BH)		Monitoring timer	TCP ULP timer value Setting time = setting value × 500 ms	3CH (60)	○	Section 5.2
12 (CH)			TCP zero window timer value Setting time = setting value × 500 ms	14H (20)		
13 (DH)			TCP resend timer value Setting time = setting value × 500 ms	14H (20)		
14 (EH)			TCP end timer value Setting time = setting value × 500 ms	28H (40)		
15 (FH)			IP assembly timer value Setting time = setting value × 500 ms	AH (10)		
16 (10H)			Response monitoring timer value Setting time = setting value × 500 ms	3CH (60)		
17 (11H)			Destination existence confirmation starting interval Setting time = setting value × 500 ms	4B0H (1200)		
18 (12H)			Destination existence confirmation interval timer Setting time = setting value × 500 ms	14H (20)		
19 (13H)			Destination existence confirmation resend timer	3H (3)		
20 (14H)		Automatic open UDP port number		1388H	×	Section 5.8
21 to 30 (15 to 1EH)		System area		–	–	–

(Continues on the next page)

*1 Setting from GX Developer allowed/prohibited

○: Setting allowed ×: Setting prohibited

*2 "Application" in the reference section column refers to the User's Manual (Application).

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name		Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
30 (1E$_H$)	Initial processing parameter setting area (For reinitialization)	TCP Maximum Segment Transmission setting area	TCP Maximum Segment Transmission setting area 0_H : Enable TCP Maximum Segment Size Option transmission 8000_H: Disable TCP Maximum Segment Size Option transmission • Re-initialization makes the set value effective.	8000_H	×	Section 5.2.3
31 (1F$_H$)		Communication condition setting area (Operational Settings)	Communication condition setting (Operational Settings) area • Communication data code setting (b1) 0: Communication in binary code 1: Communication in ASCII code • TCP Existence confirmation setting (b4) 0: Use the Ping 1: Use the KeepAlive • Send frame setting (b5) 0: Ethernet frame 1: IEEE 802.3 frame • Setting of write enable/disable at RUN time (b6) 0: Disable 1: Enable • Initial timing setting (b8) 0: Do not wait for OPEN (communication impossible at STOP time) 1: Always wait for OPEN (communication possible at STOP time) • Re-initial specification (b15) 0: Re-initial processing complete (reset by the system) 1: Re-initial processing request (set by the user) Bits other than above are reserved for system use.	0_H	○	
32 (20$_H$)	Communication parameter setting area	Connection usage setting area	Connection No. 1 • Usage of fixed buffer (b0) 0: For sending or fixed buffer communication is not executed 1: For receiving • Destination existence confirmation (b1) 0: No confirm 1: Confirm • Pairing open (b7) 0: No pairs 1: Pairs • Communication method (protocol) (b8) 0: TCP/IP 1: UDP/IP • Fixed buffer communication (b9) 0: Procedure exist 1: No procedure • Open system (b15, b14) 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open Bits other than above are reserved for system use.	0_H	○	Section 5.5
33 (21$_H$)			Connection No. 2 (bit configuration is the same as connection No.1)	(Same as connection No.1)		
34 (22$_H$)			Connection No.3 (bit configuration is the same as connection No.1)	(Same as connection No.1)		
35 (23$_H$)			Connection No.4 (bit configuration is the same as connection No.1)	(Same as connection No.1)		

(Continues on the next page)

3 - 14

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
36 (24H)	Communication parameters setting area	Communication address setting area	Connection No.5 (bit configuration is the same as connection No.1)		(Same as connection No.1)		
37 (25H)			Connection No.6 (bit configuration is the same as connection No.1)		(Same as connection No.1)		
38 (26H)			Connection No.7 (bit configuration is the same as connection No.1)		(Same as connection No.1)		
39 (27H)			Connection No.8 (bit configuration is the same as connection No.1)		(Same as connection No.1)		
40 (28H)			Connection No. 1	Local station Port No.	0H	○	Section 5.5
41 to 42 (29 to 2AH)				Destination IP address	0H	○	Section 5.5
43 (2BH)				Destination Port No.	0H	○	Section 5.5
44 to 46 (2C to 2EH)				Destination Ethernet address	FFFFFFFFFFFFH	×	—
47 to 53 (2F to 35H)			Connection No. 2	(Same as connection No. 1)			
54 to 60 (36 to 3CH)			Connection No. 3	(Same as connection No. 1)			
61 to 67 (3D to 43H)			Connection No. 4	(Same as connection No. 1)			
68 to 74 (44 to 4AH)			Connection No. 5	(Same as connection No. 1)			
75 to 81 (4B to 51H)			Connection No. 6	(Same as connection No. 1)			
82 to 88 (52 to 58H)			Connection No. 7	(Same as connection No. 1)			
89 to 95 (59 to 5FH)			Connection No. 8	(Same as connection No. 1)			
96 to 102 (60 to 66H)		System area			—	—	—
103 to 104 (67 to 68H)	Communication status storage area	System area			—	—	—
105 (69H)		Area for initial processing	Initial error code		0H	×	Section 11.3
106 to 107 (6A to 6BH)			Local station IP address		0H	×	—
108 to 110 (6C to 6EH)			Local station Ethernet address		0H	×	—
111 to 115 (6F to 73H)			System area		—	×	—
116 (74H)			Automatic open UDP port number		0H	×	—
117 (75H)			System area		—	—	—
118 (76H)			Local station network number/station number		0H	×	—
119 (77H)			Local station group number		0H	×	—

(Continues on the next page)

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name				Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
120 (78H)	Communication status storage area	Connection information area	Connection No. 1	Local station Port No.		0H	×	—
121 to 122 (79 to 7AH)				Destination IP address		0H	×	—
123 (7BH)				Destination Port No.		0H	×	—
124 (7CH)				Open error code		0H	×	Section 11.3
125 (7DH)				Fixed buffer sending error code		0H	×	Section 11.3
126 (7EH)				Connection end code		0H	×	Section 11.3
127 (7FH)				Fixed buffer communication time	Maximum value	0H	×	—
128 (80H)					Minimum value	0H	×	—
129 (81H)					Current value	0H	×	—
130 to 139 (82 to 8BH)			Connection No. 2	(Same as connection No. 1)				
140 to 149 (8C to 95H)			Connection No. 3	(Same as connection No. 1)				
150 to 159 (96 to 9FH)			Connection No. 4	(Same as connection No. 1)				
160 to 169 (A0 to A9H)			Connection No. 5	(Same as connection No. 1)				
170 to 179 (AA to B3H)			Connection No. 6	(Same as connection No. 1)				
180 to 189 (B4 to BDH)			Connection No. 7	(Same as connection No. 1)				
190 to 199 (BE to C7H)			Connection No. 8	(Same as connection No. 1)				
200 (C8H)		Area for module status	LED on/off status (Stores the on/off status of the LEDs on the front of the Ethernet module) • [INIT.] LED (b0) 0: off 1: on (initial processing completed) • [OPEN] LED (b1) 0: off 1: on (connection open processing completed) • [ERR.] LED (b3) 0: off 1: on (setting error) • [COM.ERR.] LED (b4) 0: off 1: on (communication error) Bits other than above are reserved for system use.			0H	×	—
201 (C9H)			Hub connection status area • Communication mode (b9) 0: Half duplex 1: Full duplex • Hub connection status (b10) 0: Hub not connected/disconnected 1: Hub connected • Data transmission speed (b14) 0: Operating at 10BASE-T 1: Operating at 100BASE-TX Bits other than above are reserved for system use.			0H	×	Section 5.10
202 (CAH)			• Switch status (operational mode setting) 0: Online 1: Offline 2: Self refrain test 3: Hardware test (H/W test)			0H	○	Section 4.6

(Continues on the next page)

3 - 16

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name		Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
203 (CB_H)	Communication status storage area	Module status area	Status of settings with GX Developer • Communication data code setting (b1) 0: Communication in binary code 1: Communication in ASCII code • Initial/open method setting (b2) 0: No parameter setting (start up according to the sequence program) 1: Parameter setting (start up according to the parameters) • TCP Existence confirmation setting (b4) 0: Use the Ping 1: Use the KeepAlive • Send frame setting (b5) 0: Ethernet frame 1: IEEE802.3 frame • Setting of write enable/disable at RUN time (b6) 0: Disable 1: Enable • Initial timing setting (b8) 0: Do not wait for OPEN (Communications impossible at STOP time) 1: Always wait for OPEN (Communication possible at STOP time) Bits other than above are reserved for system use.		0_H	○	Section 4.7
204 (CC_H)		Area for sending/ receiving instructions	System area	—	—	—
205 (CD_H)			RECV instruction execution request	0_H	×	Chapter 4 of Application
206 (CE_H)			System area	—	—	—
207 (CF_H)			Data link instruction execution per channel: Execution result of ZNRD instruction	0_H	×	Chapter 4 of Application
208 ($D0_H$)			Data link instruction execution per channel: System area	—	—	—
209 ($D1_H$)			Data link instruction execution per channel: Execution result of ZNWR instruction	0_H	×	Chapter 4 of Application
210 to 223 (D2 to DF_H)			Data link instruction execution per channel: System area	—	—	—

(Continues on the next page)

3 - 17

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
224 to 226 (E0 to E2H)	Error log area	System area			—	—	—
227 (E3H)		Number of error occurrence			0H	×	Section 11.3
228 (E4H)		Error log write pointer					
229 (E5H)		Error log block area	Error log block 1	Error code/end code	0H	×	Section 11.3
230 (E6H)				Subheader			
231 (E7H)				Command code			
232 (E8H)				Connection No.			
233 (E9H)				Local station Port No.			
234 to 235 (EA to EBH)				Destination IP address			
236 (ECH)				Destination Port No.			
237 (EDH)				System area	—	—	—
238 to 246 (EE to F6H)			Error log block 2	(Same as error log block 1)			
247 to 255 (F7 to FFH)			Error log block 3	(Same as error log block 1)			
256 to 264 (100 to 108H)			Error log block 4	(Same as error log block 1)			
265 to 273 (109 to 111H)			Error log block 5	(Same as error log block 1)			
274 to 282 (112 to 11AH)			Error log block 6	(Same as error log block 1)			
283 to 291 (11B to 123H)			Error log block 7	(Same as error log block 1)			
292 to 300 (124 to 12CH)			Error log block 8	(Same as error log block 1)			
301 to 309 (12D to 135H)			Error log block 9	(Same as error log block 1)			
310 to 318 (136 to 13EH)			Error log block 10	(Same as error log block 1)			
319 to 327 (13F to 147H)			Error log block 11	(Same as error log block 1)			
328 to 336 (148 to 150H)			Error log block 12	(Same as error log block 1)			
337 to 345 (151 to 159H)			Error log block 13	(Same as error log block 1)			
346 to 354 (15A to 162H)			Error log block 14	(Same as error log block 1)			
355 to 363 (163 to 16BH)			Error log block 15	(Same as error log block 1)			
364 to 372 (16C to 174H)			Error log block 16	(Same as error log block 1)			
373 to 375 (175 to 177H)		System area			—	—	—

(Continues on the next page)

3 - 18

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
376 to 377 (178 to 179H)	Error log area	Status for each protocol	IP	Received IP packet count	0H	×	Section 11.3
378 to 379 (17A to 17BH)				Received IP packet count discarded due to sum check error			
380 to 381 (17C to 17DH)				Sent IP packet total count			
382 to 397 (17E to 18DH)				System area	—	—	—
398 to 399 (18E to 18FH)				Simultaneous transmission error detection count	0H	×	Section 11.4 POINT (3)
400 to 407 (190 to 197H)				System area	—	—	—
408 to 409 (198 to 199H)			ICMP	Received ICMP packet count	0H	×	
410 to 411 (19A to 19BH)				Received ICMP packet count discarded due to sum check error			
412 to 413 (19C to 19DH)				Sent ICMP packet total count			
414 to 415 (19E to 19FH)				Echo request total count of received ICMP packets			
416 to 417 (1A0 to 1A1H)				Echo reply total count of sent ICMP packets			
418 to 419 (1A2 to 1A3H)				Echo request total count of sent ICMP packets			
420 to 421 (1A4 to 1A5H)				Echo reply total count of received ICMP packets			
422 to 439 (1A6 to 1B7H)				System area	—	—	
440 to 441 (1B8 to 1B9H)			TCP	Received TCP packet count	0H	×	—
442 to 443 (1BA to 1BBH)				Received TCP packet count discarded due to sum check error			
444 to 445 (1BC to 1BDH)				Sent TCP packet total count			
446 to 471 (1BE to 1D7H)				System area	—	—	
472 to 473 (1D8 to 1D9H)			UDP	Received UDP packet count	0H	×	
474 to 475 (1DA to 1DBH)				Received UDP packet count discarded due to sum check error			
476 to 477 (1DC to 1DDH)				Sent UDP packet total count			
478 to 481 (1DE to 1E1H)				System area	—	—	
482 to 491 (1E2 to 1EBH)			System area		—	—	
492 to 493 (1EC to 1EDH)			Receiving error	Framing error count	0H	×	
494 to 495 (1EE to 1EFH)				Overflow count	0H		
496 to 497 (1F0 to 1F1H)				crc error count	0H		
498 to 511 (1F2 to 1FFH)			System area		—	—	

(Continues on the next page)

3 - 19

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name		Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
512 to 513 (200 to 201H)	Router relay parameter setting area	Sub-net mask		0H	○	Section 5.3
514 to 515 (202 to 203H)		Default router IP address				
516 (204H)		Number of registered routers				
517 to 518 (205 to 206H)		Router 1	Sub-net address	0H	○	Section 5.3
519 to 520 (207 to 208H)			Router IP address			
521 to 524 (209 to 20CH)		Router 2	(Same as router 1)			
525 to 528 (20D to 210H)		Router 3	(Same as router 1)			
529 to 532 (211 to 214H)		Router 4	(Same as router 1)			
533 to 536 (215 to 218H)		Router 5	(Same as router 1)			
537 to 540 (219 to 21CH)		Router 6	(Same as router 1)			
541 to 544 (21D to 220H)		Router 7	(Same as router 1)			
545 to 548 (221 to 224H)		Router 8	(Same as router 1)			
549 (225H)		System area		–	–	–
550 to 551 (226 to 227H)	Station No. <-> IP information setting area	System area		–	–	–
552 (228H)		Number of conversion table data		0H	○	–
553 to 554 (229 to 22AH)		Conversion information number 1	Communication request destination/source stations network number and station number	0H	○	Chapter 3 of Application
555 to 556 (22B to 22CH)			External station Ethernet module IP address			
557 to 558 (22D to 22EH)			System area	–	–	–
559 to 564 (22F to 234H)		Conversion information number 2	(Same as conversion information No. 1)			
to		to				
931 to 936 (3A3 to 3A8H)		Conversion information number 64	(Same as conversion information No. 1)			
937 to 938 (3A9 to 3AAH)		Net mask pattern for MELSECNET/H, MELSECNET/10 routing		0H	○	Chapter 3 of Application
939 to 943 (3AB to 3AFH)		System area		–	–	–
944 to 949 (3B0 to 3B5H)	FTP setting area	FTP login name		"QJ71E71"	○	Chapter 3 of Application
950 to 953 (3B6 to 3B9H)		Password		"QJ71E71"		
954 (3BAH)		Command input monitoring timer		708H		
955 (3BBH)		PLC CPU monitoring timer		AH		
956 to 1663 (3BC to 67FH)		System area		–	–	–

(Continues on the next page)

3 - 20

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name		Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
1664 (680H)	Fixed buffer data area	Fixed buffer No. 1	Data length	0H	×	Chapter 7, Chapter 8
1665 to 2687 (681 to A7FH)			Fixed buffer data			
2688 to 3711 (A80 to E7FH)		Fixed buffer No. 2	(Same as fixed buffer No. 1)			
3712 to 4735 (E80 to 127FH)		Fixed buffer No. 3	(Same as fixed buffer No. 1)			
4736 to 5759 (1280 to 167FH)		Fixed buffer No. 4	(Same as fixed buffer No. 1)			
5760 to 6783 (1680 to 1A7FH)		Fixed buffer No. 5	(Same as fixed buffer No. 1)			
6784 to 7807 (1A80 to 1E7FH)		Fixed buffer No. 6	(Same as fixed buffer No. 1)			
7808 to 8831 (1E80 to 227FH)		Fixed buffer No. 7	(Same as fixed buffer No. 1)			
8832 to 9855 (2280 to 267FH)		Fixed buffer No. 8	(Same as fixed buffer No. 1)			
9856 to 16383 (2680 to 3FFFH)	Shared area for random access buffers and e-mail buffers	Shared area for random access buffers and e-mail buffers 1) When communicating with the random access buffer, refer to Chapter 9. 2) When using the E-mail function, refer to Chapter 2 in the User's Manual (Application).		0H	×	See the left
16384 to 20479 (4000 to 4FFFH)		System area		—	—	—
20480 (5000H)	Connection status storage area	Connection status information area	Open complete signal 0: Open incomplete 1: Open completed • Connection No. 1 (b0) • Connection No. 2 (b1) to • Connection No. 16 (b15)	0H	×	Section 5.6
20481 (5001H)			System area	—	—	—
20482 (5002H)			Open request signal 0: No open request 1: Open being requested • Connection No. 1 (b0) • Connection No. 2 (b1) to • Connection No. 16 (b15)	0H	×	Section 5.6
20483 to 20484 (5003 to 5004H)			System area	—	—	—
20485 (5005H)		Fixed buffer information area	Fixed buffer reception status signal 0: Data not received 1: Data being received • Connection No. 1 (b0) • Connection No. 2 (b1) to • Connection No. 16 (b15)	0H	×	Chapter 7
20486 (5006H)		Remote password status storage area	Remote password status 0: Unlock status/no remote password setting 1: Lock status • Connection No. 1 (b0) • Connection No. 2 (b1) to • Connection No. 16 (b15)	0H	×	Section 5.9

(Continues on the next page)

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name		Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
20487 (5007H)	System port information area	Remote password status storage area	Remote password status 0: Unlock status/no remote password setting 1: Lock status • Automatic open UDP port (b0) • GX Developer (UDP port) (b1) • GX Developer (TCP port) (b2) • FTP port (b3)	0H	×	Section 5.9
20488 (5008H)		System port use prohibited designation area	System port use prohibited designation 0: Use allowed 1: Use prohibited • Automatic open UDP port (b0) • GX Developer (UDP port) (b1) • GX Developer (TCP port) (b2)	0H	×	
20489 to 20591 (5009 to 506FH)	Monitoring area	System area		—	—	—
20592 (5070H)		Remote password function monitoring area	Remote password mismatch notification accumulated count designation (For user open port) 0: No designation 1 or higher: Notification accumulated count	1H	×	Section 5.9.6
20593 (5071H)			Remote password mismatch notification accumulated count designation (For automatic open UDP port, GX Developer communication port (TCP, UDP) and FTP communication port) 0: No designation 1 or higher: Notification accumulated count	2H		
20594 (5072H)			Connection No.1: Accumulated count of unlock process normal completion	0H		
20595 (5073H)			Connection No.1: Accumulated count of unlock process abnormal completion			
20596 (5074H)			Connection No.1: Accumulated count of lock process normal completion			
20597 (5075H)			Connection No.1: Accumulated count of lock process abnormal completion			
20598 (5076H)			Connection No.1: Accumulated count of lock process based on close			
20599 to 20603 (5077H to 507BH)			Connection No.2	(Same as connection No.1)		
20604 to 20608 (507CH to 5080H)			Connection No.3	(Same as connection No.1)		
20609 to 20613 (5081H to 5085H)			Connection No.4	(Same as connection No.1)		
20614 to 20618 (5086H to 508AH)			Connection No.5	(Same as connection No.1)		
20619 to 20623 (508BH to 508FH)			Connection No.6	(Same as connection No.1)		
20624 to 20628 (5090H to 5094H)			Connection No.7	(Same as connection No.1)		
20629 to 20633 (5095H to 5099H)			Connection No.8	(Same as connection No.1)		
20634 to 20638 (509AH to 509EH)			Connection No.9	(Same as connection No.1)		
20639 to 20643 (509FH to 50A3H)			Connection No.10	(Same as connection No.1)		
20644 to 20648 (50A4H to 50A8H)			Connection No.11	(Same as connection No.1)		

(Continues on the next page)

3 - 22

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
20649 to 20653 (50A9H to 50ADH)	Monitoring area	Remote password function monitoring area	Connection No.12	(Same as connection No.1)			
20654 to 20658 (50AEH to 50B2H)			Connection No.13	(Same as connection No.1)			
20659 to 20663 (50B3H to 50B7H)			Connection No.14	(Same as connection No.1)			
20664 to 20668 (50B8H to 50BCH)			Connection No.15	(Same as connection No.1)			
20669 to 20673 (50BDH to 50C1H)			Connection No.16	(Same as connection No.1)			
20674 to 20678 (50C2H to 50C6H)			Automatic open UDP port	(Same as connection No.1)			
20679 to 20683 (50C7H to 50CBH)			GX Developer communication UDP port	(Same as connection No.1)			
20684 to 20688 (50CCH to 50D0H)			GX Developer communication TCP port	(Same as connection No.1)			
20689 to 20693 (50D1H to 50D5H)			FTP communication port	(Same as connection No.1)			
20694 to 20736 (50D6 H to 5100H)	Status storage area	System area			—	—	—
20737 (5101H)		Error log pointer			0 H	×	Section 11.3
20738 (5102H)		Log counter (HTTP response code 100 to 199)					
20739 (5103H)		Log counter (HTTP response code 200 to 299)					
20740 (5104H)		Log counter (HTTP response code 300 to 399)					
20741 (5105H)		Log counter (HTTP response code 400 to 499)					
20742 (5106H)		Log counter (HTTP response code 500 to 599)					
20743 (5107H)		System area			—	—	—
20744 (5108H)		Error log block 1	HTTP response code		0 H	×	Section 11.3
20745 to 20746 (5109 to 510AH)			Destination IP address				
20747 to 20750 (510B to 510EH)			Error time				
20751 to 20757 (510F to 5115H)		Error log block 2	(Same as the error log block 1)				
20758 to 20764 (5116 to 511CH)		Error log block 3	(Same as the error log block 1)				
20765 to 20771 (511D to 5123H)		Error log block 4	(Same as the error log block 1)				
20772 to 20778 (5124 to 512AH)		Error log block 5	(Same as the error log block 1)				
20779 to 20785 (512B to 5131H)		Error log block 6	(Same as the error log block 1)				
20786 to 20792 (5132 to 5138H)		Error log block 7	(Same as the error log block 1)				

(Continues on the next page)

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
20793 to 20799 (5139 to 513F$_H$)	Status storage area	Error log block 8	(Same as the error log block 1)				
20800 to 20806 (5140 to 5146$_H$)		Error log block 9	(Same as the error log block 1)				
20807 to 20813 (5147 to 514D$_H$)		Error log block 10	(Same as the error log block 1)				
20814 to 20820 (514E to 5154$_H$)		Error log block 11	(Same as the error log block 1)				
20821 to 20827 (5155 to 515B$_H$)		Error log block 12	(Same as the error log block 1)				
20828 to 20834 (515C to 5162$_H$)		Error log block 13	(Same as the error log block 1)				
20835 to 20841 (5163 to 5169$_H$)		Error log block 14	(Same as the error log block 1)				
20842 to 20848 (516A to 5170$_H$)		Error log block 15	(Same as the error log block 1)				
20849 to 20855 (5171 to 5177$_H$)		Error log block 16	(Same as the error log block 1)				
20856 to 20991 (5178 to 51FF$_H$)	Use prohibited	System area			—	—	—
20992 (5200$_H$)	"Issue system switching request at disconnection detection" status storage area	"Issue system switch in cable disconnection timeout" 0: Not set 1: Set			1$_H$	○	Section 5.11
20993 (5201$_H$)		Cable disconnection timeout setting Set time = set value × 500ms (setting range: 0 to 60)			4$_H$	○	Section 5.11
20994 (5202$_H$)		System area			—	—	—
20995 (5203$_H$)		Disconnection detection count			0$_H$	×	Section 5.10
20996 to 21007 (5204 to 520F$_H$)	Use prohibited	System area			—	—	—
21008 (5210$_H$)	"System switching settings when communication error occurs" status storage area	"System switching settings when communication error occurs" (Connection for user) 0: Not set 1: Set • Connection No. 1 (b0) • Connection No. 2 (b1) to • Connection No. 16 (b15)			0$_H$	○	Section 5.11
21009 (5211$_H$)		"System switching settings when communication error occurs" (Connection for system) 0: Not set 1: Set • Auto open UDP port settings • GX Developer (UDP port) (b1) • GX Developer (TCP port) (b2) ✻ • FTP port (b3) • HTTP port (b4) ✻ MELSOFT connection included			0$_H$	○	Section 5.11
21010 to 22559 (5212 to 581F$_H$)	Use prohibited	System area			—	—	—
22560 (5820$_H$)	Communication status storage area	Connection information area	Connection No. 9	Local station Port No.	0$_H$	×	—
22561 to 22562 (5821 to 5822$_H$)				Destination IP address	0$_H$	×	—
22563 (5823$_H$)				Destination Port No.	0$_H$	×	—
22564 (5824$_H$)				Open error code	0$_H$	×	Section 11.3

(Continues on the next page)

3 - 24

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name				Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
22565 (5825H)	Communication status storage area	Connection information area	Connection No. 9	Fixed buffer sending error code		0H	×	Section 11.3
22566 (5826H)				Connection end code		0H	×	Section 11.3
22567 (5827H)				Fixed buffer communication time	Maximum value	0H	×	—
22568 (5828H)					Minimum value	0H	×	—
22569 (5829H)					Current value	0H	×	—
22570 to 22579 (582A to 5833H)			Connection No. 10	(Same as connection No.9)				
22580 to 22589 (5834 to 583DH)			Connection No. 11	(Same as connection No.9)				
22590 to 22599 (583E to 5847H)			Connection No. 12	(Same as connection No.9)				
22600 to 22609 (5848 to 5851H)			Connection No. 13	(Same as connection No.9)				
22610 to 22619 (5852 to 585BH)			Connection No. 14	(Same as connection No.9)				
22620 to 22629 (585C to 5865H)			Connection No. 15	(Same as connection No.9)				
22630 to 22639 (5866 to 586FH)			Connection No. 16	(Same as connection No.9)				
22640 (5870H)	E-mail status storage area	Receive	Number of mails remaining on the server			0H	×	Section 11.3
22641 (5871H)			Dedicated instruction normal completion count					
22642 (5872H)			Dedicated instruction abnormal completion count					
22643 (5873H)			Normal receiving count					
22644 (5874H)			Attached file receiving count					
22645 (5875H)			Server inquiry count					
22646 (5876H)			Server communication error count					
22647 (2877H)			Error log write count					
22648 (5878H)			Receiving error log write pointer					
22649 (5879H)			Error log block 1	Error code		0H	×	Section 11.3
22650 (587AH)				Command code				
22651 to 22658 (587B to 5882H)				From				
22659 to 22662 (5883 to 5886H)				Date				
22663 to 22692 (5887 to 58A4H)				Subject				
22693 to 22736 (58A5 to 58D0H)			Error log block 2	(Same as error log block 1)				
22737 to 22780 (58D1 to 58FCH)			Error log block 3	(Same as error log block 1)				
22781 to 22824 (58FD to 5928H)			Error log block 4	(Same as error log block 1)				

(Continues on the next page)

3 - 25

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
22825 to 22868 (5929 to 5954H)	E-mail status storage area	Receive	Error log block 5	(Same as error log block 1)			
22869 to 22912 (5955 to 5980H)			Error log block 6	(Same as error log block 1)			
22913 to 22956 (5981 to 59ACH)			Error log block 7	(Same as error log block 1)			
22957 to 23000 (59AD to 59D8H)			Error log block 8	(Same as error log block 1)			
23001 to 23044 (59D9 to 5A04H)			Error log block 9	(Same as error log block 1)			
23045 to 23088 (5A05 to 5A30H)			Error log block 10	(Same as error log block 1)			
23089 to 23132 (5A31 to 5A5CH)			Error log block 11	(Same as error log block 1)			
23133 to 23176 (5A5D to 5A88H)			Error log block 12	(Same as error log block 1)			
23177 to 23220 (5A89 to 5AB4H)			Error log block 13	(Same as error log block 1)			
23221 to 23264 (5AB5 to 5AE0H)			Error log block 14	(Same as error log block 1)			
23265 to 23308 (5AE1 to 5B0CH)			Error log block 15	(Same as error log block 1)			
23309 to 23352 (5B0D to 5B38H)			Error log block 16	(Same as error log block 1)			
23353 (5B39H)		Send	Dedicated instruction normal completion count		0H	×	Section 11.3
23354 (5B3AH)			Dedicated instruction abnormal completion count				
23355 (5B3BH)			Number of mails normally completed				
23356 (5B3CH)			Attached file sending count				
23357 (5B3DH)			Sending to the server count				
23358 (5B3EH)			Number of mails abnormally completed				
23359 (5B3FH)			Error log write count				
23360 (5B40H)			Error log write pointer				
23361 (5B41H)			Error log block 1	Error code	0H	×	Section 11.3
23362 (5B42H)				Command code			
23363 to 23370 (5B43 to 5B4AH)				To			
23371 to 23374 (5B4B to 5B4EH)				Date			
23375 to 23404 (5B4F to 5B6CH)				Subject			
23405 to 23448 (5B6D to 5B98H)			Error log block 2	(Same as error log block 1)			
23449 to 23492 (5B99 to 5BC4H)			Error log block 3	(Same as error log block 1)			
23493 to 23536 (5BC5 to 5BF0H)			Error log block 4	(Same as error log block 1)			
23537 to 23580 (5BF1 to 5C1CH)			Error log block 5	(Same as error log block 1)			

(Continues on the next page)

3 - 26

(Continued from the previous page)

Address Decimal (Hexadecimal)	Application	Name			Initial value (Hexadecimal)	GX Developer setting applicability	Reference section
23581 to 23624 (5C1D to 5C48H)	E-mail status storage area	Send	Error log block 6	(Same as error log block 1)			
23625 to 23668 (5C49 to 5C74H)			Error log block 7	(Same as error log block 1)			
23669 to 23712 (5C75 to 5CA0H)			Error log block 8	(Same as error log block 1)			
23713 to 24575 (5CA1 to 5FFFH)		System area			—	—	—
24576 (6000H)	Fixed buffer data area	Fixed buffer No. 9	Data length		0H	×	—
24577 to 25599 (6001 to 63FFH)			Fixed buffer data				
25600 to 26623 (6400 to 67FFH)		Fixed buffer No. 10	(Same as fixed buffer No. 9)				
26624 to 27647 (6800 to 6BFFH)		Fixed buffer No. 11	(Same as fixed buffer No. 9)				
27648 to 28671 (6C00 to 6FFFH)		Fixed buffer No. 12	(Same as fixed buffer No. 9)				
28672 to 29695 (7000 to 73FFH)		Fixed buffer No. 13	(Same as fixed buffer No. 9)				
29696 to 30719 (7400 to 77FFH)		Fixed buffer No. 14	(Same as fixed buffer No. 9)				
30720 to 31743 (7800 to 7BFFH)		Fixed buffer No. 15	(Same as fixed buffer No. 9)				
31744 to 32767 (7C00 to 7FFFH)		Fixed buffer No. 16	(Same as fixed buffer No. 9)				

3 - 27

4 SETTINGS AND PROCEDURES PRIOR TO OPERATION

This chapter explains the settings and procedures required prior to starting the operation of the Ethernet module in a system.

4.1 Loading and Installation

This section explains precautions for Ethernet module handling from unpacking to installation, as well as the installation environment common to all modules.
For more details on the module mounting and installation, refer to the User's Manual for the PLC CPU module used.

4.1.1 Handling precautions

The following explains precautions for Ethernet module handling:

(1) Since it is made of resin, the Ethernet module case should not be dropped or subjected to any shock.

(2) Tighten the module terminal and fixing screws within the following specified clamping torque range:

Screw location	Tightening torque range
External power supply terminal screw (M2.5 screw) (＊1)	40 N•cm
Module fixing screw (usually not required) (M3 screw) (＊2)	36 to 48 N•cm

＊1 This terminal is used as an external power input terminal for supplying power to the transceiver when being connected to a 10BASE5.

＊2 The module can be easily fixed onto the base unit using the hook provided at the top of the module. However, it is recommended that the module will be secured with the dedicated installation screws if it is subject to significant vibration or shock.

DANGER

- Do not touch the terminals and connectors while the power is on.
  Doing so may result in electric shocks and malfunctions.
- Make sure to turn off all phases of the external power supply before cleaning or re-tightening the screws. Failure to do so may cause damages or malfunctions of the module.
  If the screws are loose, it may cause the module to short circuit, malfunction or fall off.
  Tightening the screws excessively may damage the screws and/or the module and cause the module to short circuit, malfunction or fall off.
- Check the safety before performing control operations to the running PLC (especially, modifications of data, programs and operation status (status control)).

CAUTION

- To install the module, securely insert the module fastening latch in the installation hole on the base unit while holding down the module installation lever on the lower part of the module.
  If the module is not installed properly, it may cause the module to malfunction, fail or fall off.
  Secure the module with screws especially when it is used in an environment where constant vibrations may occurs.
- Be careful not to let any foreign matter such as wire chips get inside the module.
  They may cause fire, as well as breakdowns and malfunctions of the module.
- Never disassemble or modify the module.
  This may cause breakdowns, malfunctions, injuries or fire.
- Before mounting or dismounting the module, make sure to switch all phases of the external power supply off.
  Failure to do so may cause the module to breakdown or malfunction.
- Tighten the terminal screws using the specified torque.
  If the terminal screws are loose, it may cause the module to short-circuit, malfunction or fall off.
  Tightening the terminal screws excessively may damage the screws and/or the module and cause the module to short-circuit, malfunction or fall off.
- Do not directly touch the conducting parts and electronic parts of the module.
  This may cause the module to malfunction or fail.
- When disposing of this product, treat it as industrial waste.
- A protective sheet is pasted on the upper part of the module in order to prevent foreign matter such as wire chips to get inside the module while wiring.
  Do not remove this protective sheet during wiring work.
  However, be sure to remove the protective sheet before operating the module to allow heat radiation during operation.

4.1.2 Installation environment

This section explains the installation environment for the PLC. When installing the PLC, the following environments must be avoided:

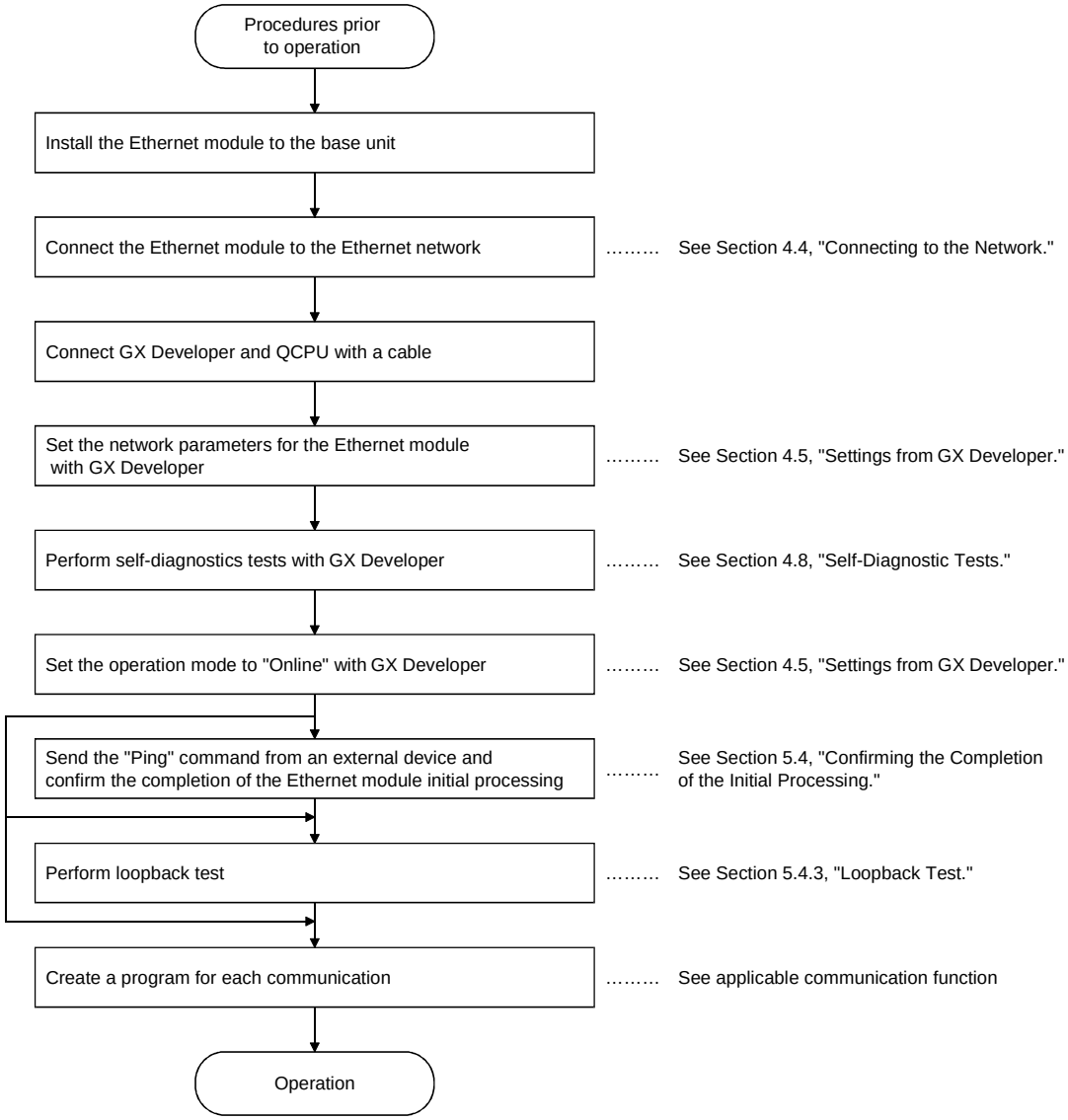
- Locations where the ambient temperature exceeds the range of 0 to 55 °C.
- Locations where the ambient humidity exceeds the range of 5 to 95 % RH.
- Locations where condensation occurs due to a sudden temperature change.
- Locations where there are corrosive or inflammable gases.
- Locations exposed to considerable amounts of conductive powdery substances such as dust and iron filing, oil mist, salt, or organic solvents.
- Locations exposed to direct sunlight.
- Location exposed to strong electric or magnetic fields.
- Location where vibrations or impacts are directly applied to the main unit.

CAUTION

- Use the PLC in the operating environment that meets the general specifications described in the user's manual of the CPU module to use.
  Using the PLC in any other operating environments may cause electric shocks, fires or malfunctions, or may damage or degrade the module.

4.2 Settings and Procedures Prior to Starting the Operation

The following shows a flow of the procedure that is required prior to starting the operation:


Procedures prior to operation
Install the Ethernet module to the base unit
Connect the Ethernet module to the Ethernet network
..... See Section 4.4, "Connecting to the Network."
Connect GX Developer and QCPU with a cable
Set the network parameters for the Ethernet module with GX Developer
..... See Section 4.5, "Settings from GX Developer."
Perform self-diagnostics tests with GX Developer
..... See Section 4.8, "Self-Diagnostic Tests."
Set the operation mode to "Online" with GX Developer
..... See Section 4.5, "Settings from GX Developer."
Send the "Ping" command from an external device and confirm the completion of the Ethernet module initial processing
..... See Section 5.4, "Confirming the Completion of the Initial Processing."
Perform loopback test
..... See Section 5.4.3, "Loopback Test."
Create a program for each communication
..... See applicable communication function
Operation

POINT

To operate the Ethernet module after the parameter settings are added or modified with GX Developer, the PLC CPU must be reset after saving the parameter values in the PLC CPU.

Important

(1) Do not write any data in the "System area" of the buffer memory of the intelligent function module.

(2) Do not output (turn on) the "Use prohibited" signal, which is one of the output signals.

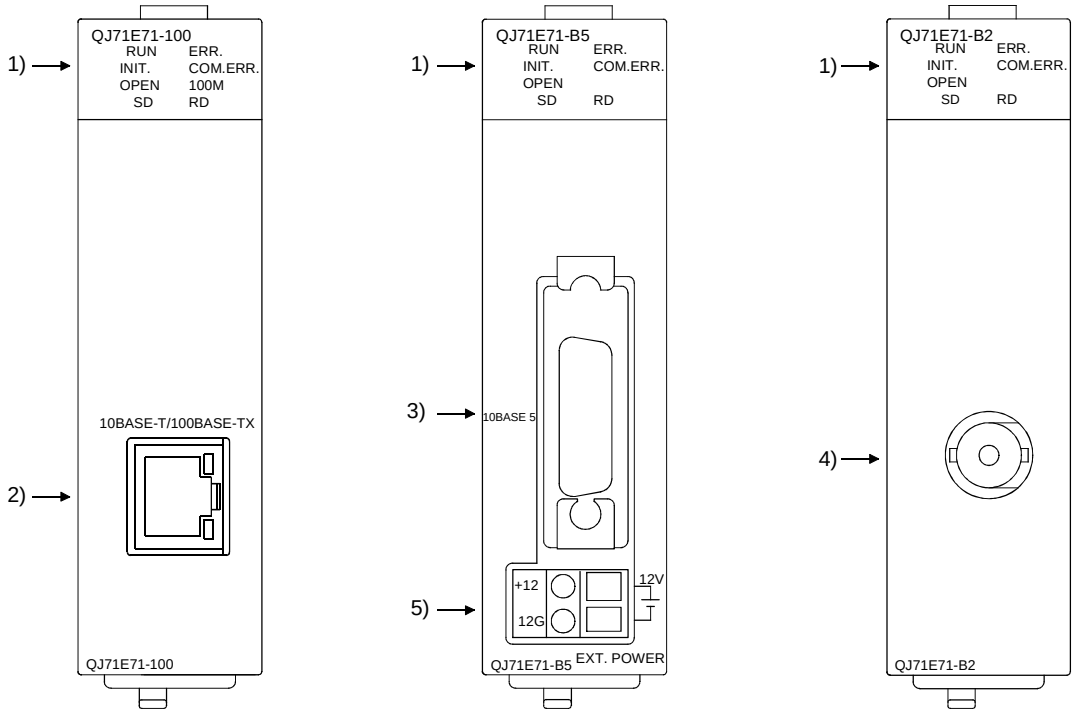
(3) When status control (such as remote RUN/STOP) from a external device is used for the PLC CPU, the user should make the selection beforehand using the selecting parameters "Always wait for OPEN". (Select using initial timing in the operation settings.)
If "Do not wait for OPEN" is selected, the communication line will be closed during remote STOP After that it cannot be reopened from the PLC CPU side and remote run from the external device also cannot be started.

(4) When using sequence programs created for Ethernet modules of conventional models, do not use the open request signals (Y8 to YF), fixed buffer communication signals (Y0 to Y7) and the dedicated OPEN/CLOSE and BUFSND/BUFRCV instructions for the same connection simultaneously in the program. It may result in malfunctioning.

(5) When the Ethernet module is replaced, reset the external device as well. (If the external device retains the Ethernet address, it may be impossible to continue the communication because a module is replaced and the Ethernet address changes.)
In the same may, when the external device (personal computer, etc.)is replaced, restart the Ethernet module.

4.3 Components of the Ethernet Module

This section shows the components of the Ethernet module.

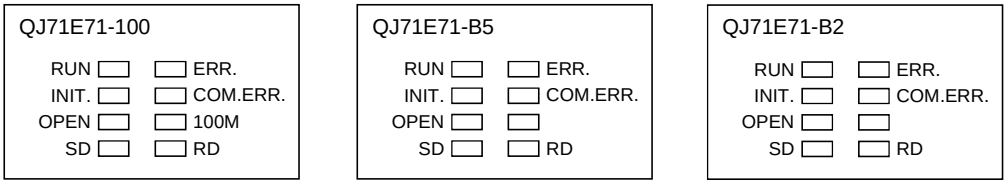


	Name	Description
1)	LED display	Refer to the contents of the LED displays (1).
2)	10BASE-T/100BASE-TX connector (RJ45) (*1)	Connector for connecting the Ethernet module to the 10BASE-T/100BASE-TX. (Ethernet module discriminate between 10BASE-T and 100BASE-TX according to the hub.)
3)	10BASE5 connector	Connector for connecting the Ethernet module to the 10BASE5 (for connecting AUI cable (transceiver cable) of 10BASE5)
4)	10BASE2 connector	Connector for connecting the Ethernet module to the 10BASE2 (for connecting 10BASE2 coaxial cable)
5)	External power supply terminal	Power supply terminal for supplying power to the transceiver when connecting by 10BASE5 (13.28 V to 15.75 V)

*1 The LED on the connector does not light up.
For connectors with first 5 digits of serial No. "05059" or earlier, the orientation of the connectors has been rotated by 180 degrees.

(1) LED display contents (*1)

1) LED display



LED name		Display description	When the LED is on	When the LED is off
QJ71E71-100	QJ71E71-B5, QJ71E71-B2			
RUN	RUN	Normal operation display	Normal	Abnormal
INIT.	INIT.	Initial processing status display	Normal completion	Not processed
OPEN *2	OPEN *2	Open processing status display	Normally opened connection available	Normally opened connection not available
SD	SD	Data sending display	Data being sent	Data is not sent
ERR.	ERR.	Setting abnormal display	Abnormal *3	Normal setting
COM.ERR.	COM.ERR.	Communication abnormal display	Communication abnormal occurrence *4	Normal communication in progress
100 M	(Not used)	Transmission speed display	100Mbps	10Mbps/When not connected
RD	RD	Data receiving status display	Data being received	Data not received

*1 Refer to Section 11.1.1 for causes of error displays and the corresponding corrective actions.

*2 The [OPEN] LED turns on/off depending on the open states of user connections 1 to 16.
(The open states of the system connections (e.g., automatic open UDP port) are not included.)

*3 The [ERR.] LED turns on in the following cases:
- When setting values in GX Developer (mode, station number, and/or network number) are incorrect.
- When an error has occurred in the Ethernet module or PLC CPU and the operation is disabled due to the error.

*4 Refer to Section 11.1.2 for the time when "COM.ERR" LED is on.

4.4 Connecting to the Network

The following explains how to connect the Ethernet module to the 100BASE-TX, 10BASE-T, 10BASE5 and 10BASE2 networks.
Some precautions that should be observed while connecting the Ethernet module are also shown below. Pay strict attention to the safety and use the Ethernet module properly.

(1) Sufficient safety precautions are required when installing the 100BASE-TX, 10BASE-T, 10BASE5 and 10BASE2 networks. Consult a specialist when connecting cable terminals or installing trunk line cables, etc.

(2) Use a connection cable conforming to the standards shown in Section 2.2.

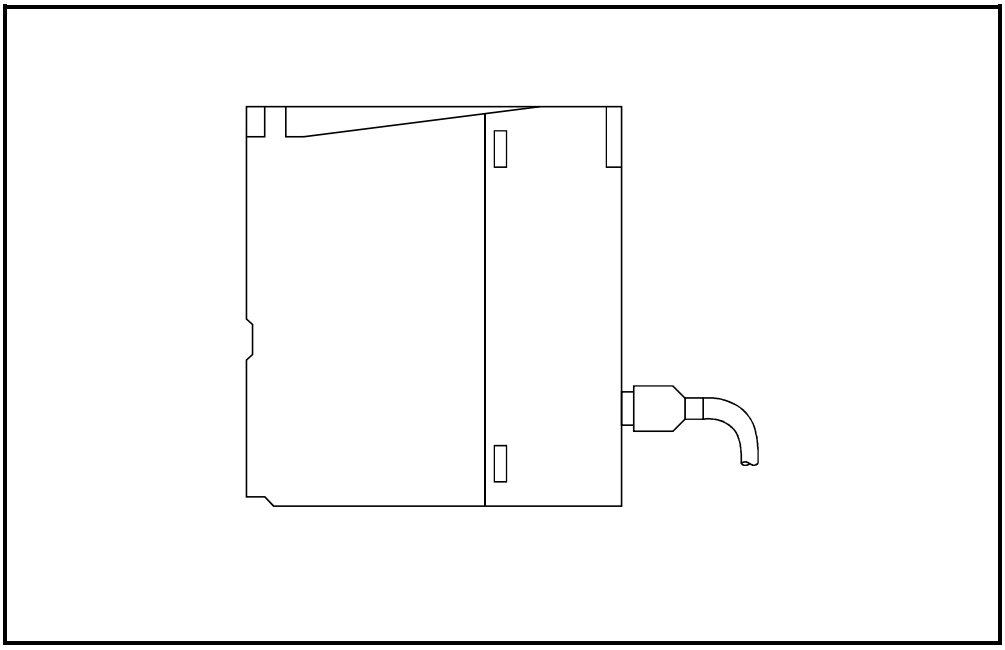
(3) The allowable bending radius for coaxial cable is preset. When bending coaxial cables to connect them, a space that is larger than the coaxial cables' allowable radius is required.
For information regarding the coaxial cables' allowable bending radius, please consult the cable manufacturer.

⚠CAUTION

- Do not bundle the control wires and the communication cables with the main circuit and the power wires, and do not install them close to each other. They should be installed at least 100 mm (3.94 in.) away from each other. Failure to do so may generate noise that may cause malfunctions.
- Do not connect the AUI cables while the power to the module installed station is on.
- Make sure to place the communication and power cables to be connected to the module in a duct or fasten them using a clamp.
  If the cables are not placed in a duct or fastened with a clamp, their positions may be unstable or moved, and they may be pulled inadvertently. This may damage the module and the cables or cause the module to malfunction because of faulty cable connections.
- When disconnecting the communication and power cables from the module, do not pull the cables by hand.
  When disconnecting a cable with a connector, hold the connector to the module by hand and pull it out to remove the cable.
  When disconnecting a cable without a connector, loosen the screws on the terminal block first before removing the cable.
  If a cable is pulled while being connected to the module, it may cause the module to malfunction or damage the module and the cable.

4.4.1 Connecting to the 10BASE-T/100BASE-TX network

This section explains how to connect the Ethernet module to the 10BASE-T, 100BASE-TX network.
(Object module for the explanation: QJ71E71-100)
The following shows the connection diagram for the twisted paid cable.



<Operating procedure>

(Step 1) Connect the twisted-pair cable to the hub.
(Step 2) Connect the twisted-pair cable to the Ethernet module.

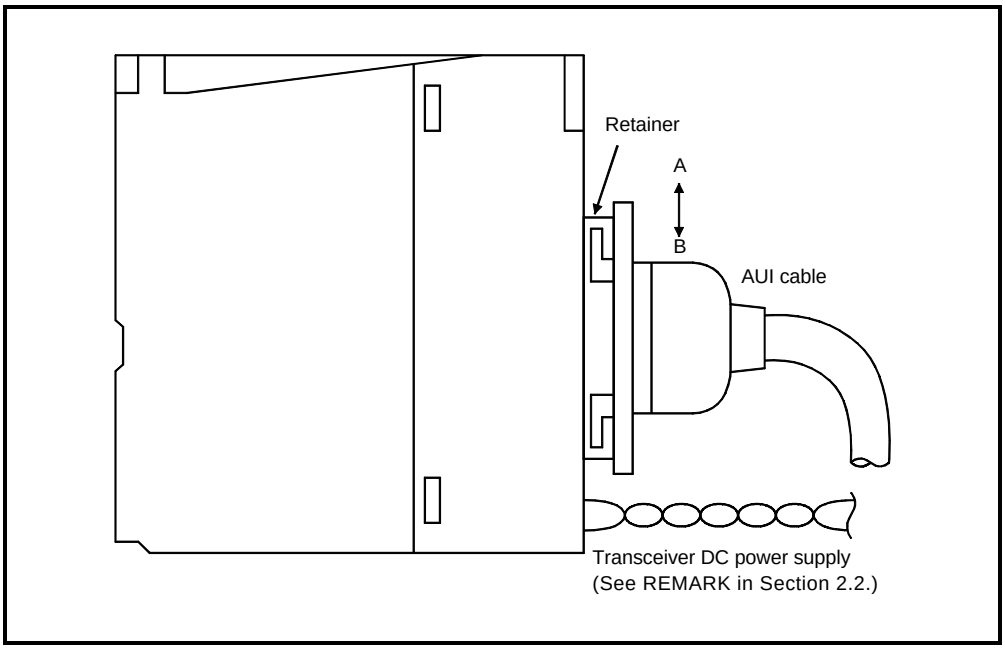
POINT

(1) The Ethernet module detects whether it is 10BASE-T or 100BASE-TX, and the full-duplex or half-duplex transmission mode according to the hub.
For connection to the hub without the auto detection function, set the half-duplex mode on the hub side.

(2) For devices required for 10BASE-T or 100BASE-TX connection and a sample system configuration, refer to Section 2.2 (1) and (2).

4.4.2 Connecting to the 10BASE5 network

This section explains how to connect the Ethernet module to the 10BASE5 network.
(Object module for the explanation: QJ71E71-B5)
The following shows the AUI cable connection diagram.



<Operating procedure>

(Step 1) Slide the retainer toward the direction B as shown above.
(Step 2) Push in the AUI cable connector all the way.
(Step 3) Slide the retainer toward the direction A as shown above.
(Step 4) Confirm that the AUI cable is locked.
(Step 5) Supply power to the transceiver (∗1).

∗1 Use a transceiver with a function that is generally called SQETEST or heart beat (a transceiver function that emits signals to notify whether the transceiver is operating normally at the end of communication).

⚠CAUTION

- Do not connect the AUI cable while the power to the module installed station is on.

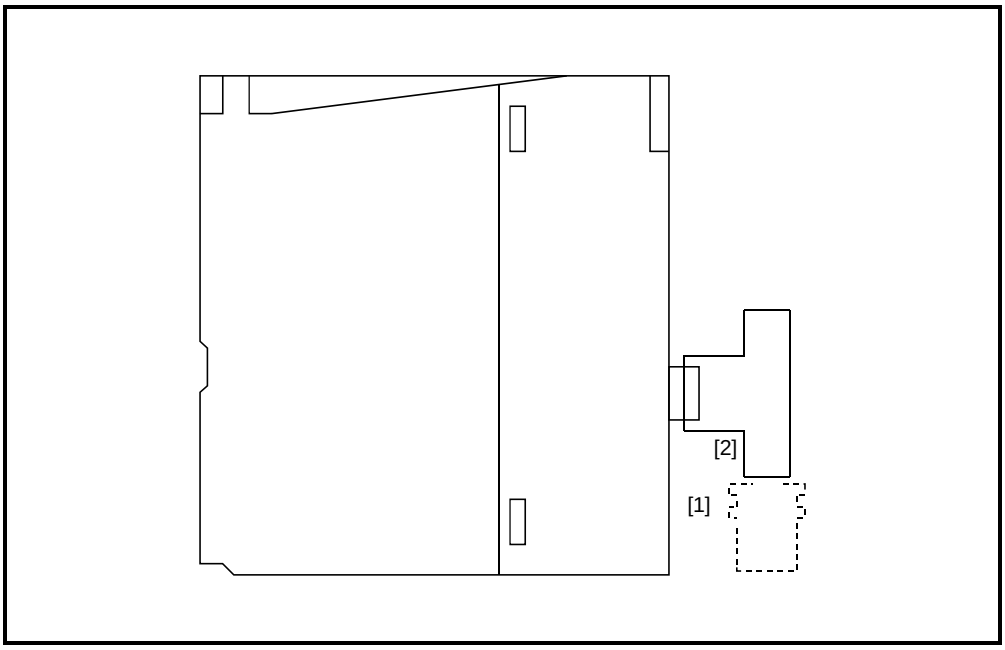
POINT

(1) When the network connection is performed using the 10BASE5 and preventing high-frequency noise is required in the installation environment of the Ethernet module, attach a ferrite core to the AUI cable to eliminate the effect of noises. See POINT in Section 2.2 a(2).

(2) See Section 2.2 (2) (b) for the device and system configration explanation that is required for 10BASE5 connection.

4.4.3 Connecting to the 10BASE2 network

This section explains how to connect the Ethernet module to the 10BASE2 network.
(Object module for the explanation: QJ71E71-B2)
The following shows the 10BASE-2 coaxial cable connection diagram.



<Operating procedure>

(Step 1) Push in the connector by aligning the groove [1] and tab [2] as shown above.
(Step 2) While pushing in the connector, rotate it clockwise a 1/4 turn.
(Step 3) Turn until the connector locks.
(Step 4) Check that the connector is locked.

POINT
See (3) in Section 2.2 for the device and system configuration example that is required for 10BASE2 connection.

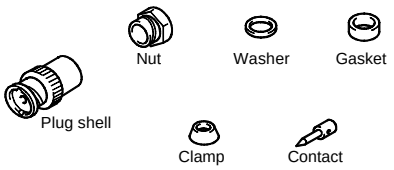
REMARK

Attaching the connector for the coaxial cable
The following section explains how to attach the BNC connector (connector plug for the coaxial cable) to the cable.

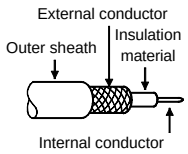
(1) Structure of the BNC connector and coaxial cable

The following shows the composition of the BNC connector and coaxial cable.

Parts comprising the BNC connector



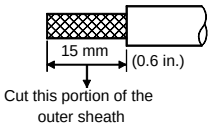
Structure of the coaxial cable



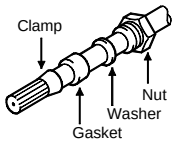
(2) How to attach the BNC connector and the coaxial cable

The following shows how to attach the BNC connector and the coaxial cable.

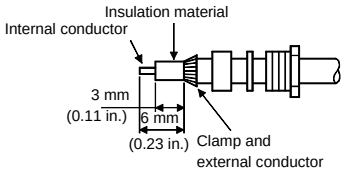
(a) Cut the length shown in the diagram to the right off the outer sheath of the coaxial cable. Be careful not to damage the external conductor.



(b) Fix the nut, washer, gasket and clamp unto the coaxial cable as shown in the diagram to the right and unfasten the external conductor.

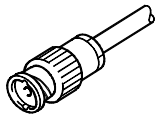


(c) Cut the external conductor, insulation material, and internal conductor to the dimensions shown in the diagram to the right. However, cut the external conductor to the same dimensions as those of the tapered Section of the clamp, and smooth it down to the clamp.



(d) Solder the contact to the internal conductor.

(e) Insert the connector assembly (d) into the plug shell and screw the nut into the plug shell.



POINT

Pay attention to the following when soldering the internal conductors and contacts:

(1) Make sure that the solder does not bead up on the soldered section.

(2) Make sure that there are no gaps between the connector and the cable insulator and that they do not cut into each other

(3) Solder as quickly as possible to prevent the insulation material from deforming.

4.5 Settings from GX Developer

This section explains the names of the setting screens for GX Developer that are available for the Ethernet module.
Perform the settings according to the function used by referring to Section 3.6, "Ethernet Module GX Developer Setting Item List."
For details on how to display each screen, refer to the GX Developer Operating Manual.

4.5.1 I/O assignment setting

[Setting purpose]

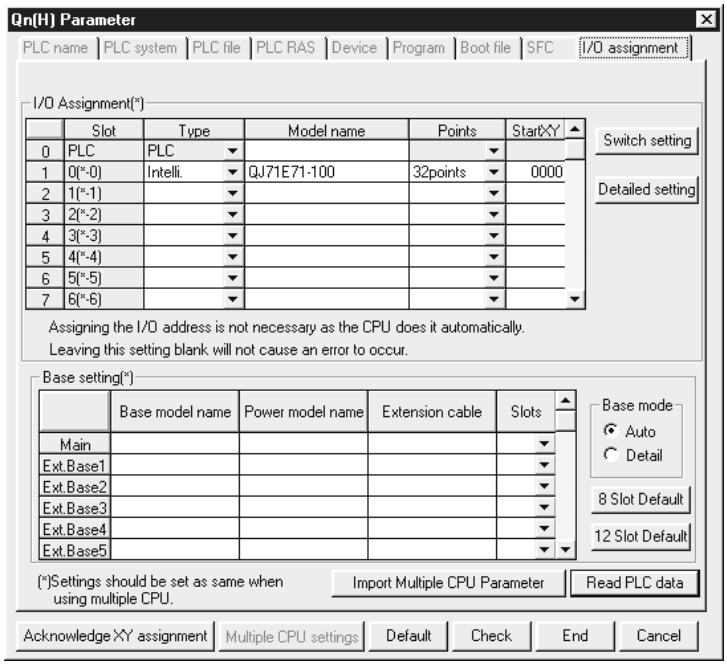
The I/O assignment settings perform the settings for the types of modules to be mounted on a base unit and the range of input/output signals.

[Startup procedure]

[GX Developer] → [PLC Parameter] → I/O assignment

For screen display, see the operating manual for GX Developer.

[Setting screen]



[Display contents]

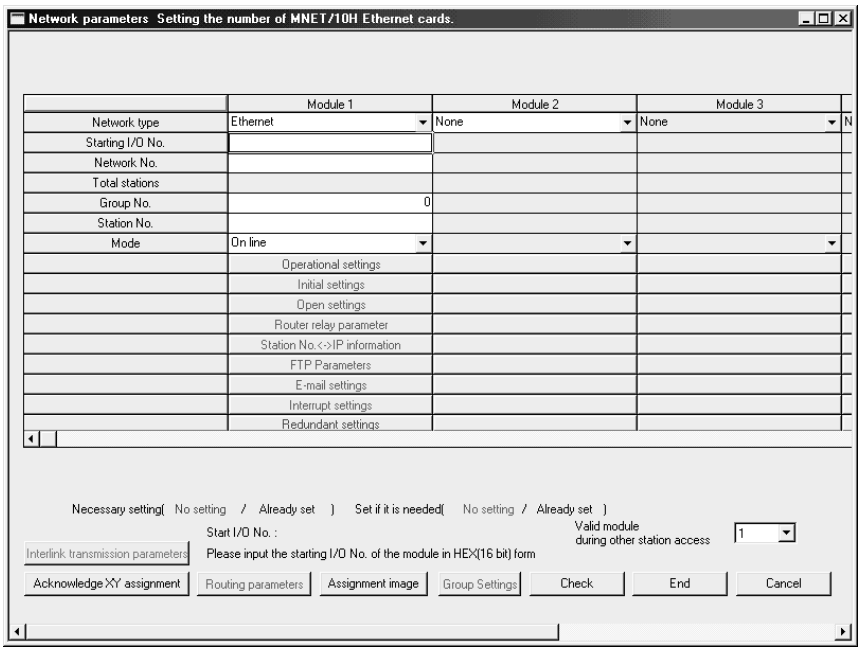
Item name		Setting for item	Remarks
I/O assignment	Type	Select "Intelli."	—
	Model name	Enter the module model name to be mounted (Example: QJ71E71-100).	
	Points	Select 32 points.	
	Start-XY	Enter the start I/O signal (hexadecimal) for the target module.	
	Detailed setting	Select the control PLC of the Ethernet module when a multiple CPU system is employed.	See QCPU User's Manual (Multiple CPU System).
Multiple CPU setting		Select when using a multiple CPU system.	

4.5.2 Other settings

This section explains the names of the setting screens for various Ethernet module functions. For setting contents, see the sections that describe respective screens in detail.

(1) "Network parameters Setting the number of MNET/10H Ethernet cards" (Details are explained in Section 4.6.)

This screen is for setting the Ethernet module as a network module.
It is also the main screen when setting the "Operational settings" and "Initial settings" to use for the Ethernet module.
The settings found on this screen must always be set in order to use the Ethernet module.



(2) "Operational settings" (Details are explained in Section 4.7.)

This screen is for setting common items for when other modules use the Ethernet module.
The settings on this screen must always be set, since they are required for the Ethernet initial processing.

(3) "Initial settings" (Details are explained in Section 5.2 and Chapter 2 of the User's Manual (Application).)

This screen is for setting common timer values for TCP/IP communication to be used in the Ethernet module as well as for setting the DNS server in order to use the e-mail function.
It is not necessary to set the timer values when communicating using the default timer values.

(4) "Open settings" (Details are explained in Section 5.5.)

This screen is for connection open processing settings and settings related to buffer memory usage when using fixed buffer communication to communicate data with an external device.

(5) "Router relay parameter (Routing information)" (Details are explained in Section 5.3.)

The following settings for data communication with external devices are performed on this screen:

- Communicating data with external devices connected to other Ethernet via a router
- Classifying each device connected to the Ethernet into groups and communicating data with an external device in an arbitrary group using the data link instructions

(6) "Station No. <-> IP information (MNET/10 routing information)" (Details are explained in Chapter 3 of the User's Manual (Application).)

This setting is for communicating with PLC CPUs on other stations via the Ethernet or MELSECNET/H, MELSECNET/10.

(7) "FTP Parameters" (Details are explained in Chapter 5 of the User's Manual (Application).)

This setting is for using the file transfer (FTP server) function.
By using the file transfer function, an external device can read/write data from/to the files in the QCPU to which the Ethernet module is installed.

(8) "E-mail settings" (Details are explained in Chapter 2 of the User's Manual (Application).)

These settings are for using the e-mail transmission/reception and automatic news (notification) functions.

(9) "Send mail address setting" (Details are explained in Chapter 2 of the User's Manual)

This screen is for setting sending destination mail addresses when using the e-mail sending/receiving function.

(10) "News settings" (Details are explained in Chapter 2 of the User's Manual (Application).)

On this screen, the settings for news (notifying) the PLC CPU monitoring results using the e-mail function are performed.

(11) "Interrupt settings" and "Interrupt pointer setting" (Details are explained in Section 7.3.)

On this screen, the settings for reading the receive data using the following functions by means of the PLC CPU interrupt program are performed.

- Reading the reception data in fixed buffer communication using a dedicated instruction (BUFRCVS).
- Reading data transmitted from another station's PLC CPU using a data link instruction (RECVS).

(12) "Redundant setting", "group setting" (Detailed explanation: Section 5.11, QCPU User's Manual)

Make settings for using the Ethernet module in a redundant system.

(13) "Routing parameters" (Details are explained in Chapter 3 of the User's Manual (Application).)

This screen is for setting which stations are passed through when communicating with PLC CPUs of other stations via the Ethernet or MELSECNET/H, MELSECNET/10.

(14) "Multiple CPU settings" (Details are explained in QCPU User's Manual.)

This screen is for setting the control PLC for the Ethernet module when it is used in a multiple CPU system.

(15) "Remote Password" (Details are explained in Section 5.9.5.)

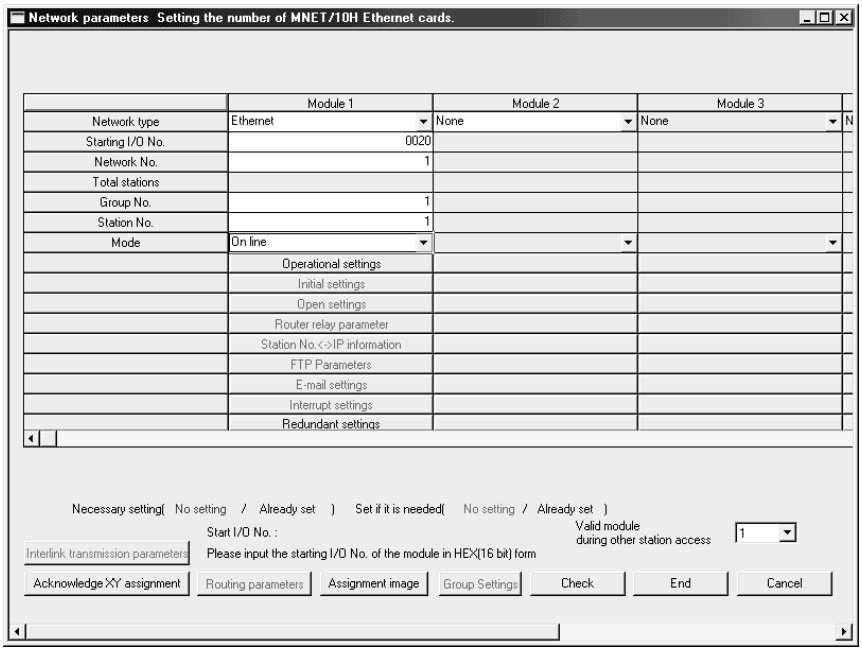
This screen is for setting connections subject to the remote password check performed by the Ethernet module when the QCPU remote password function is used.

REMARK

It is not necessary to set the "Intelligent function module switch settings" with GX Developer's I/O assignment.
Each type of setting corresponding to the switch settings is performed in the above mentioned "Operational settings," "Initial settings," and "Open settings."

4.6 Network Parameters Setting the Number of MNET/10H Ethernet Cards

This section explains one of the network parameters, setting the number of MNET/10H Ethernet cards.
Start the "Network parameters Setting the number of MNET/10H Ethernet cards" screen by selecting [GX Developer] - [Network parameter]. For details on how to display the screen, refer to the GX Developer Operating Manual.



Item name	Description of setting	Setting range/selection
Valid unit during other station access	Select the module by which the access is made when the access request does not specify the network number.	1 to 8
Network type	Select the mounting module.	Ethernet
Starting I/O No.	Set the head address of the module.	0000 to 0FE0H
Network No.	Set the network number of the module.	1 to 239
Group No.	Set the group number of the module.	1 to 32
Station No.	Select the station number of the module.	1 to 64
Mode	Select the operation mode of the module.	• Online • Offline • Self refrain test • H/W test
Operational settings	Set the common items for modules.	—
Initial settings	Set the common timer values for modules.	
Open settings	Settings for connection open processing.	
Router relay parameter	Settings for using the router relay function.	
Station No. <-> IP information	Settings for using the MELSECNET/H, MELSECNET/10 relay communication function.	
FTP Parameters	Settings for using the file transfer (FTP) function.	
E-mail settings	Settings for transmitting/receiving e-mails.	
Interrupt settings	Settings for executing an interrupt program.	
Redundant setting	Settings for using the Ethernet module in a redundant system.	
Group settings	Settings for disabling system switching if an error occurs in one communication path when the redundant communication paths are used for the external device and Ethernet module in a redundant system. (Refer to the QnPRHCPU User's Manual (Redundant System).)	
Routing parameters	Settings for using the MELSECNET/H, MELSECNET/10 relay communication function.	

(1) Valid module during other station access

(a) This item designates the network module to which the request is directed, when the local station issues a data communication request and it cannot specify the network number of the access destination PLC station. This should be one of the following:
- A network module for the MELSECNET/H, MELSECNET/10
- An Ethernet module

(2) Network type

(a) Select "Ethernet," then each setting item is displayed as shown in the above screen.
The following items must be set.
- Starting I/O No.
- Network No.
- Group No.
- Station No.
- Mode
- Operational settings
- Initial settings (Settings not required when using the default values)

(3) Starting I/O No.

(a) Set the head I/O number of the Ethernet module in units of 16 points (hexadecimal).

(b) An example of the setting is shown below.
(Example)

1)	CPU type	: Q25 (H) CPU
2)	Existing mounted module	: Serial communication module QJ71C24
3)	I/O usage status	: X/Y000 to X/Y01F (used by QJ71C24) X/Y020 to X/Y03F (used by the Ethernet module)

The head I/O number is "0020" in the above environment.

(4) Network No.

(a) Set the network number of the target Ethernet module (setting range from 1 to 239) for the MELSECNET/H, MELSECNET/10.

(b) Do not set the network number to a value already assigned to existing systems and other Ethernet and MELSECNET/H, MELSECNET/10 network systems.

(5) Group No.

(a) Set the group number of the target Ethernet module (setting range from 1 to 32) for the MELSECNET/H, MELSECNET/10.

(b) By designating a group number, data can be communicated with multiple QCPU stations using the same group number.

(6) Station No.

(a) Set the station number of the target Ethernet module (setting range from 1 to 64) for the MELSECNET/H, MELSECNET/10.

(b) Do not set the number to a value already assigned to existing systems and other Ethernet modules.

(7) Mode (address: CAH)

(a) Select the operating mode of the Ethernet module.

Name of setting	Description of setting
Online	Communicate with the external device in normal operation mode.
Offline	Disconnecting the local station from the network.
Self refrain test	Execute a self refrain test for self-diagnostics. (＊1)
H/W test	Execute tests of RAM and ROM. (＊1)

＊1 See Section 4.8, "Self-Diagnostic Tests."

(b) When the operation mode is changed, reset the PLC CPU.

(8) Operational settings through Routing parameters

Set the parameters from operational settings through routing settings according to the explanations in Section 4.5.

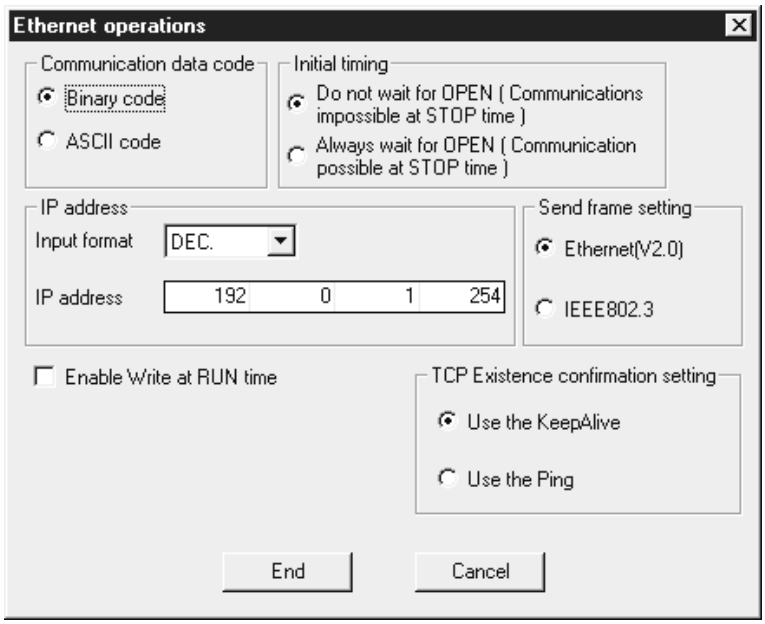
POINT

(1) Both the "Setting the number of MNET/10H Ethernet cards" and "Operational setting" parameters must always be specified. If the settings are changed, the QCPU (PLC No.1 when used in a multiple CPU system) must be reset.

(2) Write the network parameters in the control PLC of the Ethernet module when the module is used in a multiple CPU system.

4.7 Operational Settings

This section explains how to set the operations parameters.
Start the "Ethernet operations" screen by selecting [Setting the number of MNET/10H Ethernet cards] - [Operational settings].



Item name		Setting description of item	Setting range/selection
Communication data code		Select the communication data code.	• Binary code • ASCII code
Initial Timing		Perform the setting for opening.	• Do not wait for OPEN • Always wait for OPEN
IP Address	Input format	Select the IP address input format.	• Decimal • Hexadecimal
	IP address	Set the IP address of the local station.	—
Send frame setting		Select the frame format to send	• Ethernet (V2.0) • IEEE 802.3
Enable Write at RUN time (for the MC protocol)		Set to allow writing to the PLC CPU during RUN.	• Check mark (enable) • No check mark (disable)
TCP Existence confirmation setting		Select the existence check method for TCP communication	• Use the KeepAlive • Use the Ping

(1) Communication data code (address: CB_H … b1)

(a) Select the format of the communication data when communicating with an external device.

Name of setting	Description of setting
Binary code	Communicate using binary data.
ASCII code	Communicate using ASCII data.

(b) For more details on the data communication codes, see Section 3.2, "Data Codes for Communication."

4 - 19

(2) Initial Timing (address: CBH … b8)

(a) Select the timing to open for connections for which TCP-Passive open or UDP open are selected with the "Open settings" parameter (*1).

*1 For more details on the open settings, see Section 5.5 "Open Settings".

Name of setting	Description of setting
Do not wait for OPEN (Communication impossible at STOP time)	• Execute open/close processing using a sequence program. • Communication cannot be performed while the PLC CPU is in the STOP status.
Always wait for OPEN (Communication possible at STOP time)	• Passive open and UDP open connections always wait for open according to the parameter settings (a sequence program for open/close processing is not required) (*2). • Communication can be performed while the PLC CPU is in the STOP status.

*2 If the sequence program of the local station's PLC CPU executes the close processing, the station is not placed in the OPEN request wait status after the connection is shut off.

(b) In the following cases, a dedicated instruction is required to execute an open/close processing.
- When "Do not wait for OPEN" is selected in the initial timing settings.
- If the "Open settings" are not performed for a connection.
- If "TCP-Active" is selected in the "Open settings" for a connection.

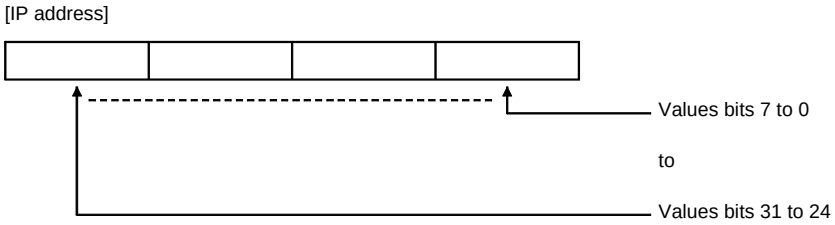
For detail on the open/close processing, see Section 5.6 "Open Processing/Close Processing of the Connection".

(3) IP Address settings – Input format

(a) Select the IP address input format (decimal or hexadecimal)

(4) IP Address settings – IP address (address: 0H to 1H)

(a) Set the IP address of the local station according to the specified input format (decimal or hexadecimal).
It should be set so that the local station Ethernet module and the communicating external device have the same class and sub-net address (two words).



(b) It is necessary to use the router relay function in order to communicate with an external device on another Ethernet (different sub-net address).
For more details, see Section 5.3, "Router Relay Parameter".

(c) Set the IP address after consulting a network administrator (the person who plans the network and manages IP addresses).

(5) Send frame setting

(a) Select the frame of the Ethernet header for the data link layer to be sent by the Ethernet module.

Setting item	Description of setting
Ethernet (V2.0)	Transmits using an Ethernet frame.
IEEE802.3	Transmits using an IEEE802.3 frame.

(b) When receiving data from the external device, reception should occur regardless of whether the Ethernet frame or IEEE802.3 frame is used

REMARK

(1) Transmission using Ethernet frames is generally recommended.
(2) When communication with the external device fails, check whether or not communication is possible using a PING command.

(6) Enable Write at RUN time (address: CBH … b6)

(a) Select enable/disable external devices to write data while communicating through the MC protocol while the PLC CPU is running.

Name of setting	Description of setting
Check mark (Enable write at RUN time)	Allows an external device to write while the PLC CPU is running.
No check mark (Disable write at RUN time)	Prohibits an external device to write while the PLC CPU is running.

(7) TCP Existence confirmation setting

Select the existence check method for TCP communication. *1
For the existence check function, refer to Section 5.2.2.

Name of setting	Description of setting
Use the KeepAlive	Checks connection status with KeepAlive.
Use the Ping	Checks connection status with Ping.

*1 Do not use GX Developer incompatible with this setting together with compatible GX Developer. (Doing so may change the setting to "Use the Ping".)
This setting is ignored when the Ethernet module does not support the KeepAlive check function. (Ping is used for the existence check.)
Refer to Section 2.7 for applicable versions of the Ethernet module and GX Developer.

POINT

(1) The "Setting the number of MNET/10H Ethernet cards" and "Operational settings" parameters must always be set.
If the settings are changed, the PLC CPU must be reset.

(2) If the re-initial processing of the Ethernet module is required due to the occurrence of an error, perform re-initial processing using sequence programs (see Section 5.2.3).

(3) When using the programs for conventional models, be sure to delete or disable the program for initial processing that uses I/O signals.

4.8 Self-Diagnostic Tests

This section explains the self-diagnostic tests for checking the hardware and transmission and reception function of the Ethernet module.
The self-diagnostic test is performed on the "Network parameters setting the number of MNET/10H Ethernet cards" screen of GX Developer.

4.8.1 Self refrain test

The following explains the self refrain test that is used to check the hardware including the Ethernet module's transmission and reception circuit.
The self refrain test transmits a test message addressed to the Ethernet module's local station to a line and checks whether or not the same message can be received via the network.
The following explains the procedure for performing the self refrain test. The test takes approximately five seconds to complete.
The test result can be determined from the LED displays on the front of the Ethernet module.

Step	Description of operation		Status of LED		
			[RUN]	[OPEN]	[ERR.]
1	Connect the Ethernet module to the line. (∗ 1) (See Section 4.4.)		—	—	—
2	Stop QCPU.		—	—	—
3	Select [Setting the Number of MNET/10H Ethernet Cards] – [Mode] to choose [self refrain test] and save the parameter values in the PLC CPU. (See Section 4.6.)		—	—	—
4	Reset QCPU. (Test starts)		●	●	○
5	Check status of each LED after 5 seconds.	Normal	●	○	○
		Error	●	○	●
6	Select [Setting the Number of MNET/10H Ethernet Cards] – [Mode], change the mode to "Online" or another test mode, and save the parameter values in the PLC CPU. (See Section 4.6.)		—	—	—
7	Reset QCPU.		—	—	—

●: Lit ○: Off

∗1 For QJ71E71-100, when the line is not connected, the self refrain test is not performed and end normally.

The following are probable causes of errors.
- Ethernet module hardware error
- Ethernet line error
- External power supply 12 V DC error (only 10BASE5)

An error code is stored in the error log area (address: E5H) in the buffer memory of the Ethernet module; the error content can then be checked from GX Developer. (See Sections 11.2 and 11.3.)

POINT

There will be no hardware interference even if the self refrain test is conducted while an external device is online. If the packets are congested over the line, packet collisions may occur and the test may not finish with the error end or within the expected 5-second time frame.
In this case, perform the test again after terminating the data communication between other devices.

4.8.2 Hardware test (H/W Test)

This section explains the RAM and ROM tests for the Ethernet module. The procedure for the hardware test is as shown in the table below.
The test results are judged from the LED displays on the front of the Ethernet module.

Step	Description of operation		Status of LED		
			[RUN]	[OPEN]	[ERR.]
1	Stop QCPU.		—	—	—
2	Select [Setting the Number of MNET/10H Ethernet Cards] – [Mode] to choose [H/W test] and save the parameter values in the PLC CPU. (See Section 4.6.)		—	—	—
3	Reset QCPU. (Test starts.)		●	●	○
4	Check status of each LED after 5 seconds.	Normal	●	○	○
		Error	●	○	●
5	Select [Setting the Number of MNET/10H Ethernet Cards] – [Mode], change the mode to "Online" or another test mode, and save the parameter values in the PLC CPU. (See Section 4.6.)		—	—	—
6	Reset QCPU.		—	—	—

●: Lit ○: Off

The following are probable causes of errors.

- Ethernet module RAM/ROM error

An error code is stored in the error log area (address: E5H) in the buffer memory of the Ethernet module; the error content can then be checked from GX Developer. (See Sections 11.2 and 11.3.)

POINT

If the result of the hardware test shows an error, conduct the test again.
If this test result also shows an error, the hardware of the Ethernet module may be faulty.
Please consult the nearest branch office or dealer with details of the errors.

4.9 Maintenance and Inspection

This section explains the maintenance and inspection as well as the mounting and dismounting of the Ethernet module.

4.9.1 Maintenance and inspection

The Ethernet module does not need to be inspected for anything particular other than checking whether or not the connections of terminators and cables are loose. Maintain and inspect the system according to the same inspection items as described in the user's manual for the PLC CPU in order to use it in optimal operating conditions.

DANGER

- Do not touch the terminals and connectors while the power is on.
  Doing so may cause in electric shocks and malfunctions.
- Do not touch the connector inside the cover at the upper section of the module.
  Doing so may cause damages and malfunctions of the module.
- Make sure to turn off all phases of the external power supply before cleaning or re-tightening the screws. Failure to do so may cause damages or malfunctions of the module.
  If the screws are loose, it may cause the module to short-circuit, malfunction or fall off.
  Tightening the screws excessively may damage the screws and/or the module and cause the module to short circuit, malfunction or fall off.

CAUTION

- Be careful not to let any foreign matter such as wire chips get inside the module.
  They may cause fire, as well as breakdowns and malfunctions of the module.
- Never disassemble or modify the module.
  This may cause breakdowns, malfunctions, injuries or fire.

4.9.2 Mounting and dismounting the module

Before mounting or dismounting the Ethernet module, make sure to read Section 4.1, "Handling Precautions" thoroughly, secure the safety, and handle the module properly according to the instructions.
The following explains the procedures when mounting/dismounting the Ethernet module.

<Operation procedure when replacing the Ethernet module>

(Step 1) Turn off the power supply to the Ethernet module installed station.
(Step 2) Remove the network cable and Ethernet module.
(Step 3) Install and start up the new Ethernet module according to Section 4.2, "Settings and Procedures Prior to Starting the Operation."
(Step 4) Reset the external device. (*1)

<Operation procedure when replacing the QCPU>

(Step 1) Read the parameters for the Ethernet module from the PLC CPU and save them using GX Developer. (*2)
(Step 2) Replace the QCPU. (See the applicable QCPU User's Manual.)
(Step 3) Load the parameters for the Ethernet module that have been saved in GX Developer to the new QCPU.
(Step 4) Reset the external device.

*1 When the Ethernet module is replaced, reset the external device as well. (If the external device retains the Ethernet address, it may be impossible to continue the communication because a module is replaced and the Ethernet address changes.)
In the same may, when the external device (personal computer, etc.)is replaced, restart the Ethernet module.

*2 It is recommended to record and save parameters not only when the CPU is replaced but also when parameters for the Ethernet module are created or modified.

5 COMMUNICATION PROCEDURE

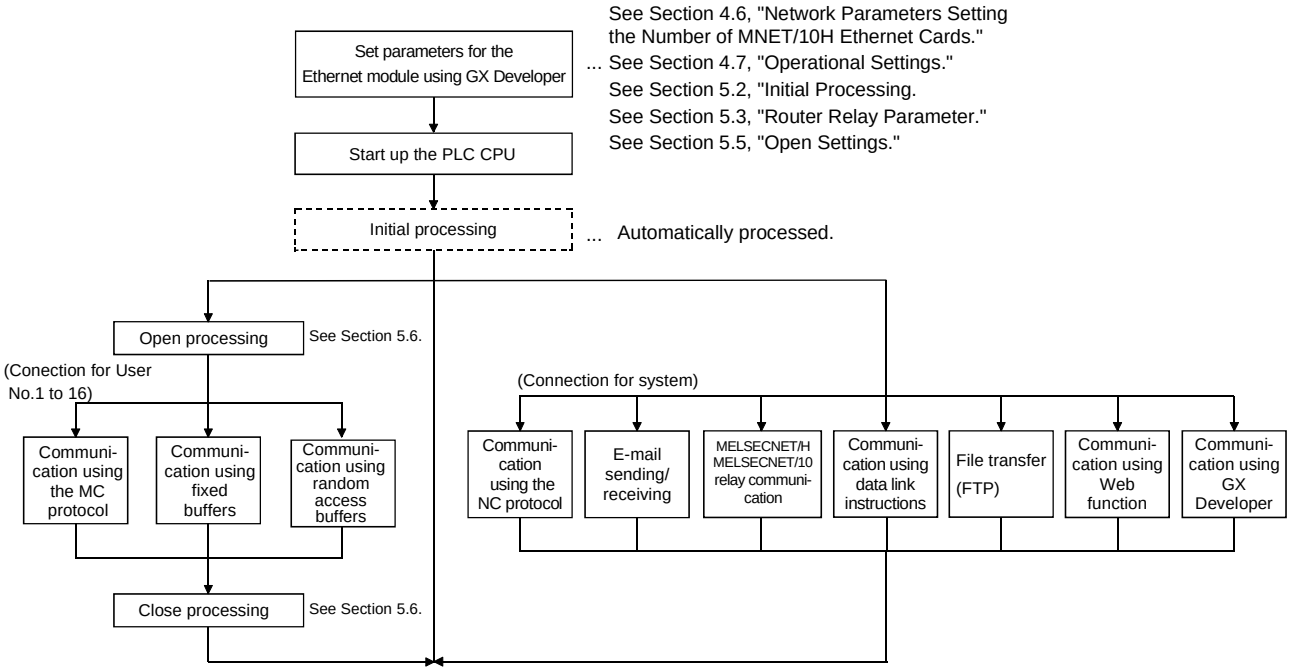
This chapter explains an overview of the communication procedure using the Ethernet module, as well as the required initial processing of the Ethernet prior to starting data communication and the open processing for communication with an external device.
The sequence program for setting the parameters required to perform communication can be greatly simplified by setting the parameters of the Ethernet module supported by GX Developer.
The following explains the communication procedure using GX Developer.

REMARK

When utilizing a sequence program for a conventional module, see Appendix, "Using Programs Designed for Conventional Modules".

5.1 Overview of the Communication Procedure

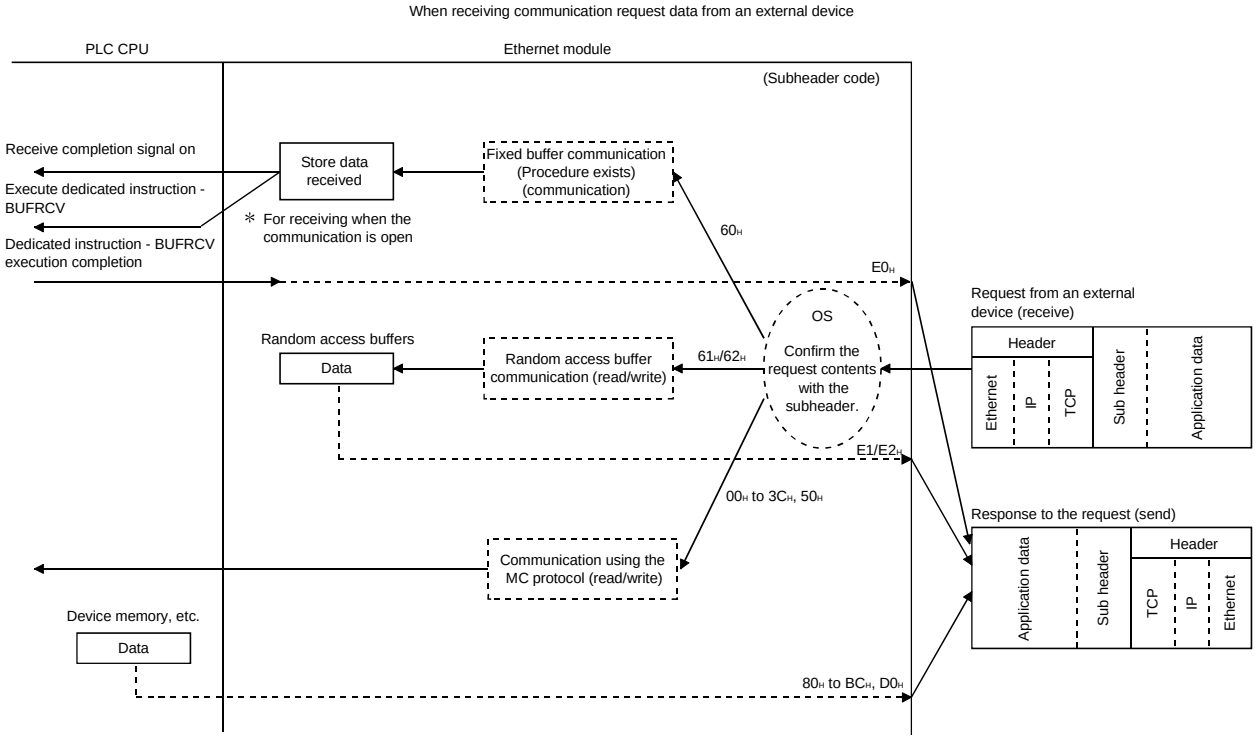
This section explains an overview of the procedure for performing data communication with external devices via the Ethernet module.
To start data communication, establish connections with external devices via the initial or open processing.
To end the data communication, perform close and end processing to shut off the connection and, as a result, all communication processing is terminated.
The following diagram illustrates the communication procedure:


Set parameters for the Ethernet module using GX Developer
... See Section 4.6, "Network Parameters Setting the Number of MNET/10H Ethernet Cards."
... See Section 4.7, "Operational Settings."
See Section 5.2, "Initial Processing.
See Section 5.3, "Router Relay Parameter."
See Section 5.5, "Open Settings."
Start up the PLC CPU
Initial processing
... Automatically processed.
Open processing
See Section 5.6.
(Conection for User No.1 to 16)
(Connection for system)
Communi-cation using the MC protocol
Communi-cation using fixed buffers
Communi-cation using random access buffers
Communi-cation using the NC protocol
E-mail sending/ receiving
MELSECNET/H MELSECNET/10 relay communi-cation
Communi-cation using data link instructions
File transfer (FTP)
Communi-cation using Web function
Communi-cation using GX Developer
Close processing
See Section 5.6.

REMARK

Each of the following three types of communication can be performed with an external device opened by the user.

- Communication using the MC protocol
- Sending/receiving in fixed buffer communication (procedure exists)
- Communication using random access buffers



5.2 Initial Processing

This section explains the initial processing of the Ethernet module.

5.2.1 Initial processing

The initial processing is for enabling data communication with an external device by setting the minimum parameters required for data communication by the Ethernet module.
Set the following parameters using GX Developer, save them in the PLC CPU, and reset the PLC CPU; the initial processing of the Ethernet module is then performed. (Sequence programs for initial processing are not required.)

(1) Parameters required for initial processing (set using GX Developer)

- "Network parameter Setting the number of Ethernet cards" : (See Section 4.6.)
- "Operational settings" : (See Section 4.7.)
- "Initial settings" : (See Section 5.2.2.)

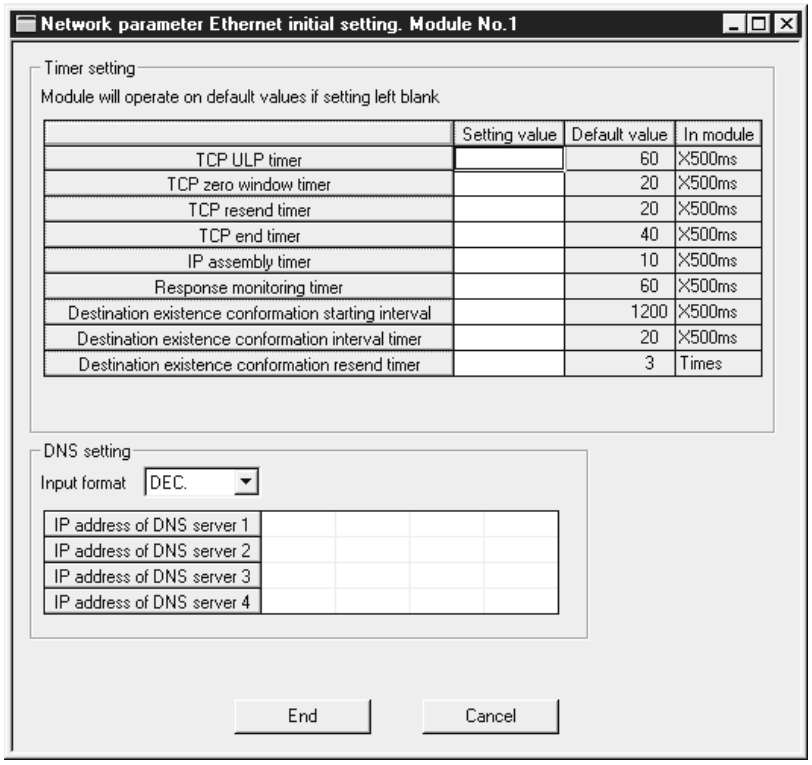
(2) Confirmation of initial processing result

Initial processing	Ethernet module		
	INIT.LED	I/O signal	
		Initial normal completion signal X19	Initial abnormal completion signal X1A
At normal completion	●: On	ON	OFF
At abnormal completion	○: Off	OFF	ON

If the initial processing does not complete normally, correct the above parameter setting value and write to the PLC CPU. Then reset the PLC CPU.

5.2.2 Initial settings

This section explains the initial settings using GX Developer.
Select [Setting the number of MNET/10H Ethernet cards] – [Initial settings] to start the [Ethernet initial settings] screen.


Network parameter Ethernet initial setting. Module No.1
Timer setting
Module will operate on default values if setting left blank.
Setting value
Default value
In module
TCP ULP timer 60 X500ms
TCP zero window timer 20 X500ms
TCP resend timer 20 X500ms
TCP end timer 40 X500ms
IP assembly timer 10 X500ms
Response monitoring timer 60 X500ms
Destination existence conformation starting interval 1200 X500ms
Destination existence conformation interval timer 20 X500ms
Destination existence conformation resend timer 3 Times
DNS setting
Input format DEC.
IP address of DNS server 1
IP address of DNS server 2
IP address of DNS server 3
IP address of DNS server 4
End
Cancel

Item name		Description of setting	Setting range/options
Timer setting	TCP ULP timer	Set the time of packet existence at TCP data transmission.	2 to 32767
	TCP zero window timer	Set the interval for checking the reception enabled status.	2 to 32767
	TCP resend timer	Set the time to resend at TCP data transmission.	2 to 32767
	TCP end timer	Set the confirmation wait time at TCP close processing.	2 to 32767
	IP assembly timer	Set the wait time for division data packets.	1 to 32766
	Response monitoring timer	Set the response wait time.	2 to 32767
	Destination existence confirmation starting interval	Set the time to start confirming existence of an external device after communication with it has terminated.	1 to 32767
	Destination existence confirmation interval timer	Set the time interval between reconfirming existence.	1 to 32767
	Destination existence confirmation resend timer	Set the number of times to reconfirm existence when a response to the existence confirmation is not received.	1 to 32767
DNS setting (＊1)	Input format	Select IP address input format of DNS server.	Decimal/hexadecimal
	IP address of DNS server 1	Set IP address of DNS server 1.	—
	IP address of DNS server 2	Set IP address of DNS server 2.	—
	IP address of DNS server 3	Set IP address of DNS server 3.	—
	IP address of DNS server 4	Set IP address of DNS server 4.	—

＊1: For details, see Chapter 2 of the User's Manual (Application).

5 - 4

(1) Timer setting – TCP ULP timer (address: BH)

(a) This item sets the time of packet existence during TCP data sending. This timer is passed through the parameter when TCP opens or data is sent.

(b) Designate the setting value in a range from 2 to 32767. (When the default value is used, setting is not required.)

(c) Timer setting = setting value × 500 ms

(2) Timer setting - TCP zero window timer (address: CH)

(a) The window indicates the reception buffer on the receiving side.

(b) When there is no more space in the receiving buffer (window size = 0) on the receiving side, data communication has to wait until enough space is made.
When this occurs, the sending side sends a sending window confirmation packet to the receiving side after the TCP zero window timer value has been reached, and confirms the reception enabled status.

(c) Designate the setting value in a range from 2 to 32767. (When the default value is used, setting is not required.)

(d) Timer setting = setting value × 500 ms

(3) Timer setting – TCP resend timer (address: DH)

(a) Set the resend time if ACK is not returned at TCP opening or data transmission. This timer is also used for the existence time of the ARP function.
(ARP is resent in TCP resend timer value/2 if a response is not returned in reply to the sent ARP request.)
It also serves as the minimum setting time for the data link instruction arrival monitoring time.

(b) Designate the setting value in a range from 2 to 32767. (When the default value is used, setting is not required.)

(c) Timer setting = setting value × 500 ms

(4) Timer setting – TCP end timer (address: EH)

(a) When the TCP connection is closed from the local station, this timer sets the monitoring time for how long the local station waits for a FIN request from an external device after it sends an FIN request and the external device returns an ACK.

(b) If the FIN request cannot be received from the external device before the time designated by the TCP end timer setting, a RST should be sent to the external device to forcefully close the connection.

(c) Designate the setting value in a range from 2 to 32767. (When the default value is used, setting is not required.)

(d) Timer setting = setting value × 500 ms

(5) Timer setting - IP assembly timer (address: FH)

(a) Communication data may be divided on the IP level due to the restriction on the buffer on the sending or receiving station.

(b) Designate the setting value in a range from 1 to 32766.
(When the default value is used, setting is not required.)

(c) Timer setting = setting value × 500 ms

(6) Timer setting - Response monitoring timer (address: 10H)

(a) This timer setting sets the following times.

1) The time to wait for a response after sending a command.

2) The time to wait for the last message after receiving the first message when the message is divided.

(b) Designate the setting value in a range from 2 to 32767.
(When the default value is used, setting is not required.)

(c) Timer setting = setting value × 500 ms

(7) Timer setting - Destination existence confirmation starting interval (address: 11H)

(a) This timer sets the time interval before starting to confirm the existence of an external side when an open connection for which existence confirmation is set to confirm stops dead.

(b) Designate the setting value in a range from 1 to 32767.
(When the default value is used, setting is not required.)

(c) Timer setting = setting value × 500 ms

(8) Timer setting - Destination existence confirmation interval timer (address: 12H)

(a) This timer sets the time interval before reconfirming the existence of an external device on an open connection for which existence confirmation is set to confirm that does not respond.

(b) Designate the setting value in a range from 1 to 32767.
(When the default value is used, setting is not required.)

(c) Timer setting = setting value × 500 ms

(9) Timer setting - Destination existence confirmation resend timer (address: 13H)

(a) This timer sets the number of times to reconfirm existence when there is no response from an external device on an open connection for which existence confirmation is set to confirm.

(b) Designate the setting value in a range from 1 to 32767.
(When the default value is used, setting is not required.)

(10) DNS setting - Input format (∗1)

(11) DNS setting - IP address of DNS server n (∗1)

∗1 The DNS setting is set when the e-mail sending/receiving function is used. Refer to Chapter 2, "E-mail Function" of the User's Manual (Application).

REMARK

(1) Designate the setting value of each timer on the Ethernet module side in such a way that the following relations are met.

- [Response monitoring timer value] ≧ [TCP ULP timer value] ≧ [TCP end timer value] ≧ [TCP resend timer value] > [IP assembly timer value]
- [TCP resend timer value] = [TCP zero window timer value]

Furthermore, when connecting a line using our products, you should make sure that both nodes have the same settings.

(2) Designate the setting value of each timer on the external device side in such a way that the following relations are met.
Communication errors such as transmission timeouts may occur more frequently if the timer values are not set so that they satisfy the following relationships.

- [TCP resend timer value on the external device side] > [TCP resend timer value on the Ethernet module side]
- [Monitoring timer value for the external device application software] > {[TCP ULP timer value on the Ethernet module side] × n ∗1}

∗1 n is the number of TCP segment transmissions and can be obtained via the following calculation:

$$n = \left\lceil \frac{\text{Size of the message transmitted by the Ethernet module}}{\text{Maximum segment size}} \right\rceil$$ (fractions below decimal point are rounded up)

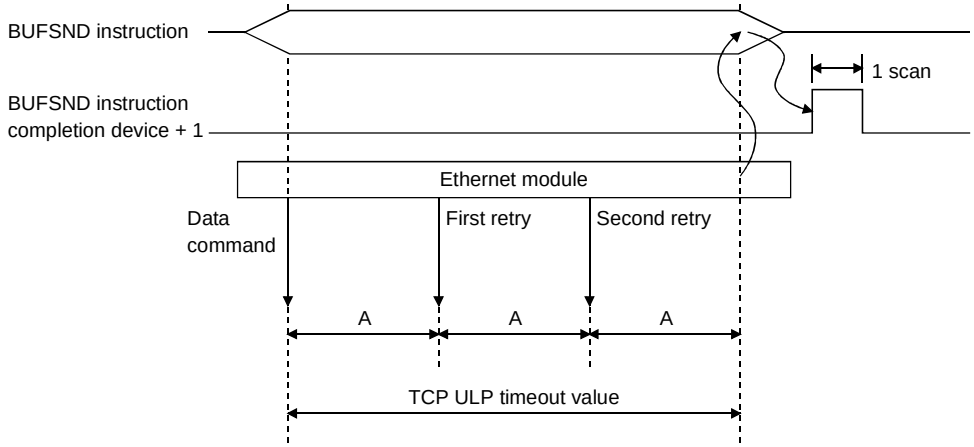
Example1: Number of TCP segment transmissions when communicating via the same line
The maximum segment size is 1460 bytes via the same line (without going through a router) and the number of TCP segment transmissions is as follows:
n = 1, if the size of the message transmitted by the Ethernet module is 1460 bytes or less.
n = 2, if the size of the message transmitted by the Ethernet module is greater than 1460 bytes.

Example2: Number of TCP segment transmissions when communicating via separate lines
The maximum segment size is at least 536 bytes on a separate line (via dialup router, etc.) and the number of TCP segment transmissions is as follows:
n = 1, if the size of the message transmitted by the Ethernet module is 536 bytes or less.
n = 2, if the size of the message transmitted by the Ethernet module is greater than 536 bytes and no more than 1072 bytes.
n = 3, if the size of the message transmitted by the Ethernet module is greater than 1072 bytes and no more than 1608 bytes.

(3) When communication errors occur due to noise or other factors, the settings should be changed to increase the number of retries.
The number of retries is obtained by using the following equation:
(In case of the default values, 2 = (60/20) - 1)

- $\left(\text{Number of retries}\right) = \left\{\left(\frac{\text{TCP ULP timer value}}{\text{TCP resend timer value}}\right) - 1\right\}$

Example: Assuming the values are set in such a way that the number of retries is two, a data transmission error will occur at the timing shown in the figure below if data transmission fails (when communicating using fixed buffer).



A: TCP resend timer value
(The time after which the data should be retransmitted when an "ACK" is not returned after sending data.)

(4) Perform the following setting in order to eliminate the retries explained in (3) (i.e., to set the number of retries to 0).

- $\left(\text{TCP ULP timer value}\right) = \left(\text{TCP end timer value}\right) = \left(\text{TCP resend timer value}\right)$

(Each timer value should be identical.)

(5) The target existence check is a function whereby the Ethernet module checks whether or not a remote device is functioning normally by sending an existence check message and then waiting to see whether a response message is received. It is used if a connection to a remote device is open but communication with the remote device has not been performed for a certain period of time.

(a) The existence check function has two methods of checking: PING and KeepAlive.
The Ethernet module performs each of the existence checks based on the setting values explained in (7) to (9) of this section and the existence check setting of the open settings (refer to Section 5.5 (6)).
The existence check function (Ping or KeepAlive) can be selected at the time of operation setting or re-initialization.
For the operation setting, refer to Section 4.7.
Refer to Section 5.2.3 for the explanation about the re-initial processing.

1) Checking by KeepAlive

This method is used for a connection opened via the TCP/IP protocol. The Ethernet module performs an existence check by sending an existence check ACK message to a remote device with which communication has not been performed for a certain period of time and waiting to see whether or not a response is received. (*1)

*1 The connection may be cut off if the remote device does not support the TCP KeepAlive function (response to KeepAlive ACK messages).

2) Checking by PING

This method is used for a connection opened via the TCP/IP or UDP/IP protocol.

The Ethernet module performs an existence check by sending a PING command (using the ICMP echo request/response function) to a remote device with which communication has not been performed for a certain period of time and waiting to see whether or not a response is received. (*2)

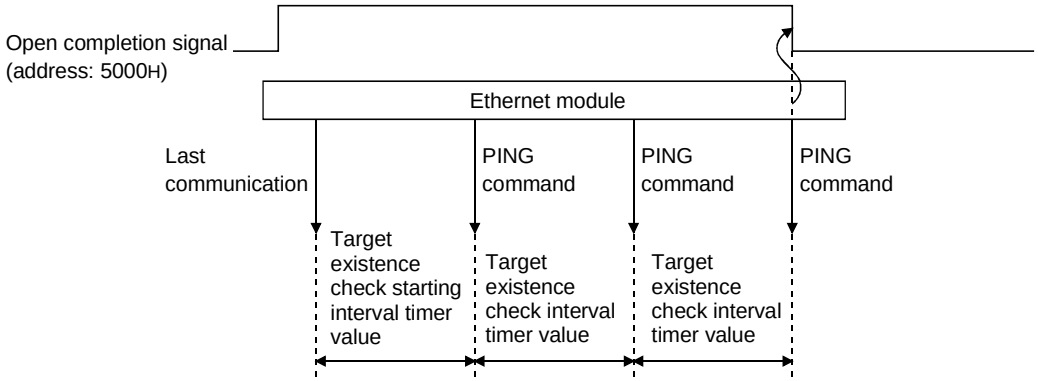
*2 Note that the Ethernet module automatically returns an echo response packet when it receives a PING echo request command. (It sends a response to the received PING command even if the connection used in the data communication with the remote device is closed.)

(b) The following actions are taken if a response message cannot be received (an error is detected) from the remote device.

- The corresponding connection will be forcibly closed (the line is disconnected). (*3)

  *3 It is possible to reopen it with a user program.
- The Ethernet module turns off the open completion signal (the corresponding bit in address: 5000H) and stores the error code (C035H) in places such as the open error code storage area.

Example: Assuming the values are set under the condition that the number of retries is three, the Ethernet module performs target existence check at the timing shown in the figure below. (An example of existence check by PING)



5.2.3 Re-initial Processing

Re-initial processing is performed in order to place the Ethernet module into its startup status without actually restarting the PLC (e.g., resetting the PLC CPU).
Re-initial processing of the Ethernet module can be performed in a sequence program. The purposes of and how to program the re-initial processing of the Ethernet module are explained below.

(1) Purposes of performing re-initial processing

(a) To update address information of an external device maintained by the Ethernet module

The Ethernet module maintains the IP address of the external device with which it has been communicating and the corresponding Ethernet address (MAC address). This is done in order to prevent other devices from accessing the PLC illegally using the IP address of an external device with which communication was performed normally. (∗1)

For this reason it is necessary to perform a re-initial processing in order to clear the address information of the external device maintained by the Ethernet module in case a module or board on the external device side has been replaced due to failure.

∗1 Ethernet address is unique to a device. No devices share the same Ethernet address.

(b) To change the IP address of the Ethernet module of the local station

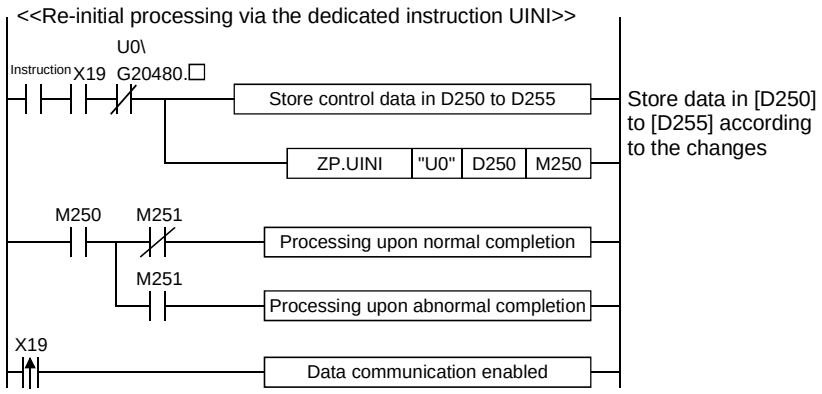
If the system is changed, it is possible to restart communication with external devices by changing the IP address (for the Ethernet module of the local station) in the operation settings (see Section 4.7) set by GX Developer.

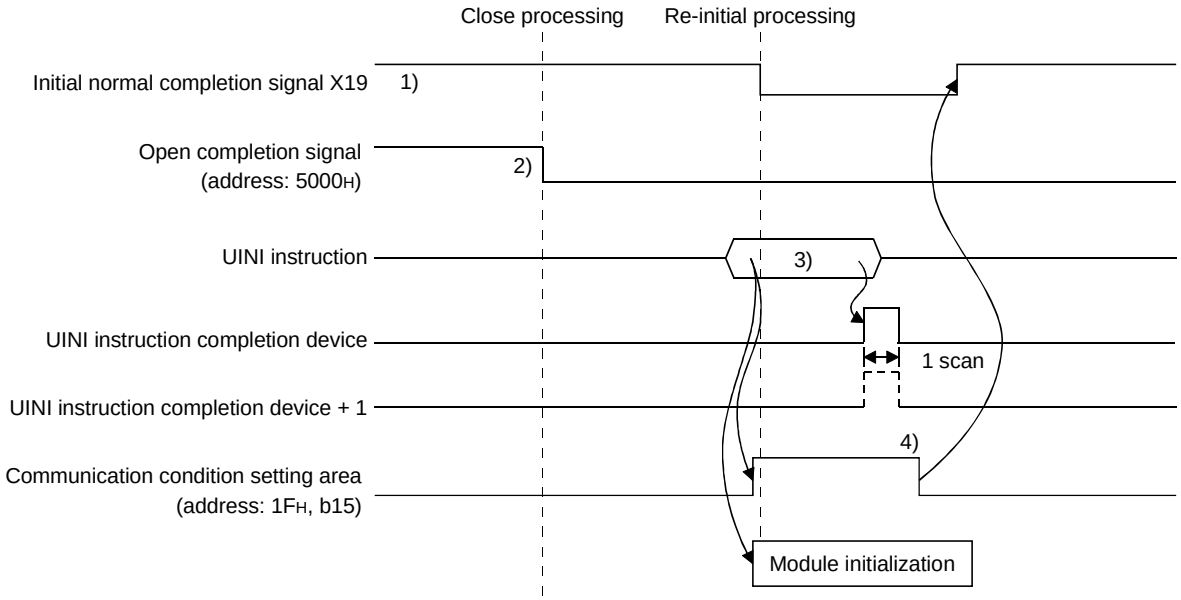
(c) To change the communication condition setting values

It is possible to restart communication with external devices by changing the communication conditions in the operation settings (see Section 4.7) set by GX Developer.

(2) Programming and timing for performing a re-initial processing

The programming and timing for performing a re-initial processing is shown below:





1) It is confirmed whether the initial processing has been completed normally.

2) All data communication currently being performed with external devices is terminated, and close processing will be performed on all connections.

3) Dedicated instruction UINI is executed.
(Initial normal completion signal (X19): ON)
(Open completion signal (address: 5000H (20480)): All OFF (0H))
The Ethernet module is initialized after the parameters (local station IP address, operation settings) have been specified via the control data of the dedicated instruction.

4) When the re-initial processing is completed, the re-initial specification (address: 1FH, bit 15) becomes "0" and the initial processing completion signal X19 turns on.
∗ If the re-initial processing is completed abnormally, the error code is stored in the following area.
Initial error code storage area (address: 69H (105))

(3) Precautions for combination with the MELSOFT products when the setting is changed to "Enable TCP Maximum Segment Size Option transmission" in re-initial processing

(a) MELSOFT series products supporting the TCP Maximum Segment transmission function

For setting the "Enable TCP Maximum Segment Size Option transmission", use the following MELSOFT products.

GX Developer	: Version8.07H or later
MX Component	: Version3.03D or later
MX Links	: Version3.08J or later

(b) MELSOFT series products not listed above (a)

When using a MELSOFT product that is not listed above (a) to make communication via Ethernet, set it to "Disable TCP Maximum Segment Size Option transmission" or use UDP/IP type communication.
Otherwise, the sequence program may not function correctly (read/write is incorrect).

POINT

Please keep the following points in mind when re-initializing the Ethernet module. (Failure to do so may cause errors in the data communication with the external devices.)

(1) Be sure to end all current data communication with external devices and close all connections before performing a re-initial processing.

(2) Do not mix a re-initial processing done by writing directly into buffer memory, for instance by using a TO instruction, with a re-initial processing via UINI instruction. Also, do not request another re-initial processing while a re-initial processing is already being performed.

(3) Be sure to reset external devices if the IP address of the Ethernet module has been changed. (If an external device maintains the Ethernet address of a device with which it communicates, the communication may not be continued after the IP address of the Ethernet module has been changed.)

REMARK

It is possible to change the operational settings when performing re-initial processing by a sequence program.

In the case of Ethernet module, parameters other than the operational settings, the settings should be changed via GX Developer, after which the QCPU (PLC No. 1 in the case of a multiple CPU system) should be reset.

Parameter setting item			Applicable buffer memory address (hexadecimal)	Parameter changes	Reference section
PLC parameters			—	×	—
	I/O assignment settings		—	×	Section 4.5
	Interrupt pointer settings		—	×	Section 7.3
Network parameters Setting the number of MNET/10H Ethernet cards			—	×	Section 4.6
	Operational settings		1 F_H	○	Section 4.7, 5.2.3
	Initial settings	Timer setting	B_H to 13_H	△ (＊1)	Section 5.2
		DNS settings	—	×	User's Manual (Application), Chapter 2
	Open settings		20_H to $5F_H$	△ (＊1)	Section 5.5
	Router relay parameter		4_H, 200_H to 224_H	△ (＊1)	Section 5.3
	Station No. <-> IP information		4_H, 229_H to $3A9_H$	△ (＊1)	User's Manual (Application), Chapter 3
	FTP parameters		4_H, $3B0_H$ to $3BB_H$	△ (＊1)	User's Manual (Application), Chapter 5
	E-mail settings		—	×	User's Manual (Application), Chapter 2
		Send mail address setting	—	×	
		News setting	—	×	
	Interrupt settings		—	×	Section 7.3
	Redundant setting		—	×	Section 5.11.3
	Routing parameters		—	×	User's Manual (Application), Chapter 3
Remote password settings			—	×	Section 5.9.5

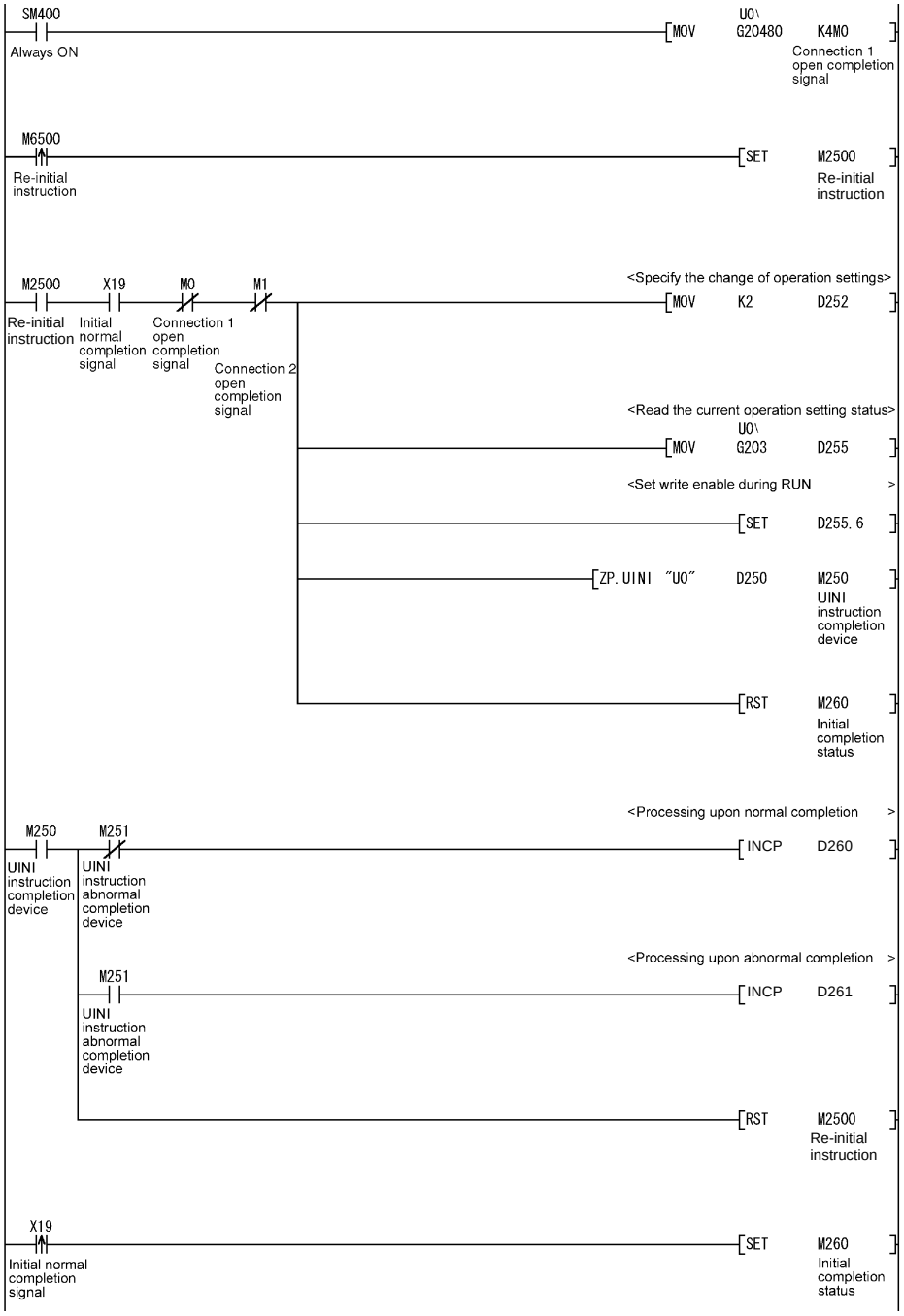
○△ : Valid, × : Invalid

*1 The PLC CPU operates with the settings performed during the re-initial processing in the applicable buffer memories. Do not change the settings in the applicable buffer memories.

(4) Sample Program

The following figure shows a sample program that performs a re-initial processing.

When I/O signals of the Ethernet module are X/Y00 to X/Y1F



POINT

This is an example program that can be used to perform re-initial processing when communicating with connections 1 and 2. The corresponding numbers and bits for each connection should be specified if other connections are used.

REMARK

Direct write to the following buffer memory addresses enables the re-initial processing of the Ethernet module.
The procedure for performing the re-initial processing by directly writing to the buffer memory is explained below:

(a) Change the value stored in the TCP Maximum Segment transmission area (address: 1EH).
0000H: Enable TCP Maximum Segment Size Option transmission
8000H: Disable TCP Maximum Segment Size Option transmission

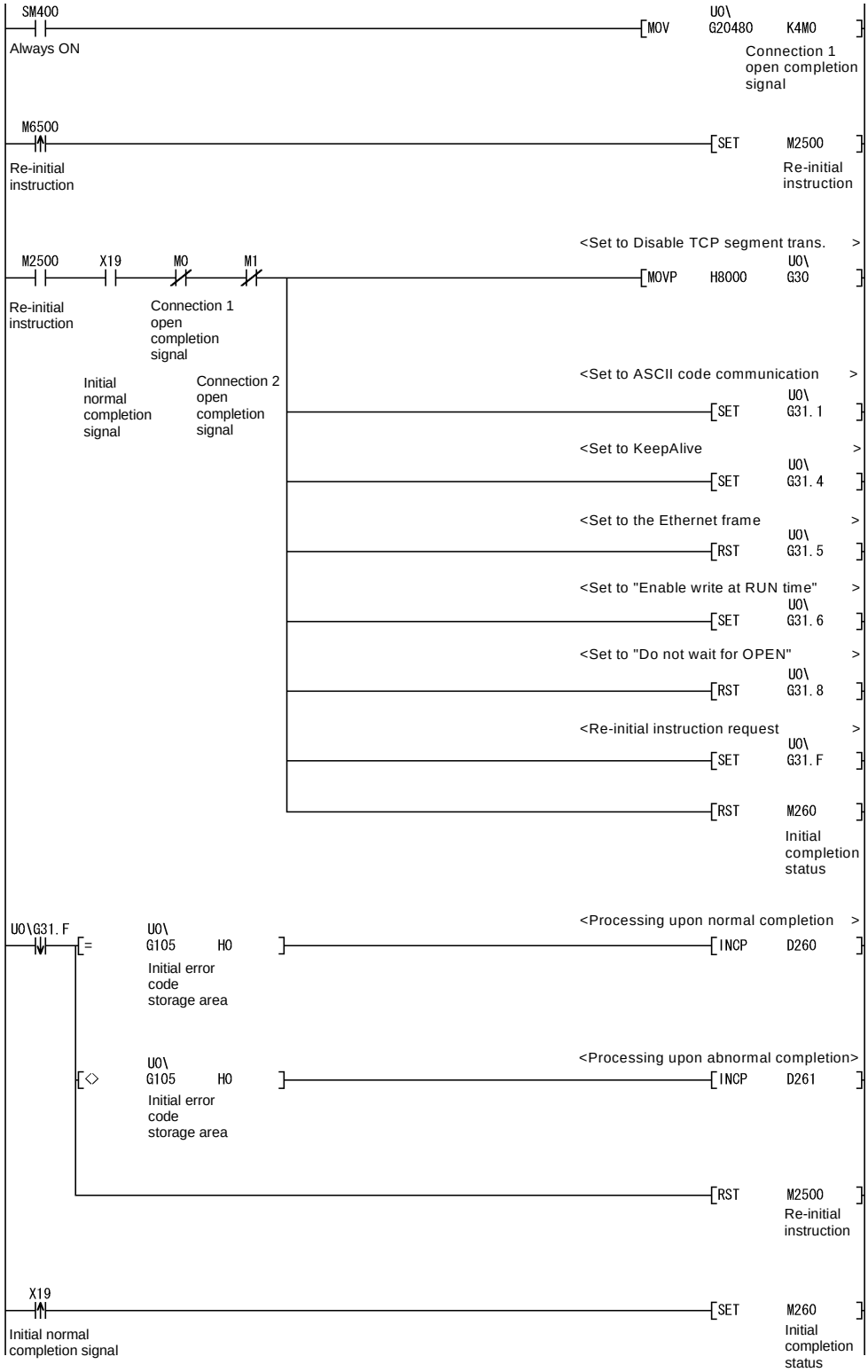
(b) Change the values stored in the communication condition setting area (address: 1FH).
Bits 0 to 14 : Specify the values set in the operation settings.
Bit 15 : "1"

	b15			b8		b6	b5	b4		b1	
Buffer memory address 1FH	6)	0		5)	0	4)	3)	2)		1)	0

1) Communication data code setting (b1)
0: Communication in binary code
1: Communication in ASCII code
2) TCP Existence confirmation setting (b4)
0: Use the Ping
1: Use the KeepAlive
3) Send frame setting (b5)
0: Ethernet frame
1: IEEE802.3 frame
4) Setting of write enable/disable at RUN time (b6)
0: Disable
1: Enable
5) Initial timing setting (b8)
0: Do not wait for OPEN (communication impossible at STOP time)
1: Always wait for OPEN (communication possible at STOP time)
6) Re-initial specification (b15)
0: Re-initial processing complete (reset by the system)
1: Re-initial processing request (set by the user)

(c) Once the re-initial processing is complete, the re-initial specification (address: 1FH, bit 15) is set to "0." The initial processing completion signal X19 also turns ON.
∗ If the re-initial processing completed abnormally, an error code will be stored in the following area:
Initial error code storage area (address: 69H(105))

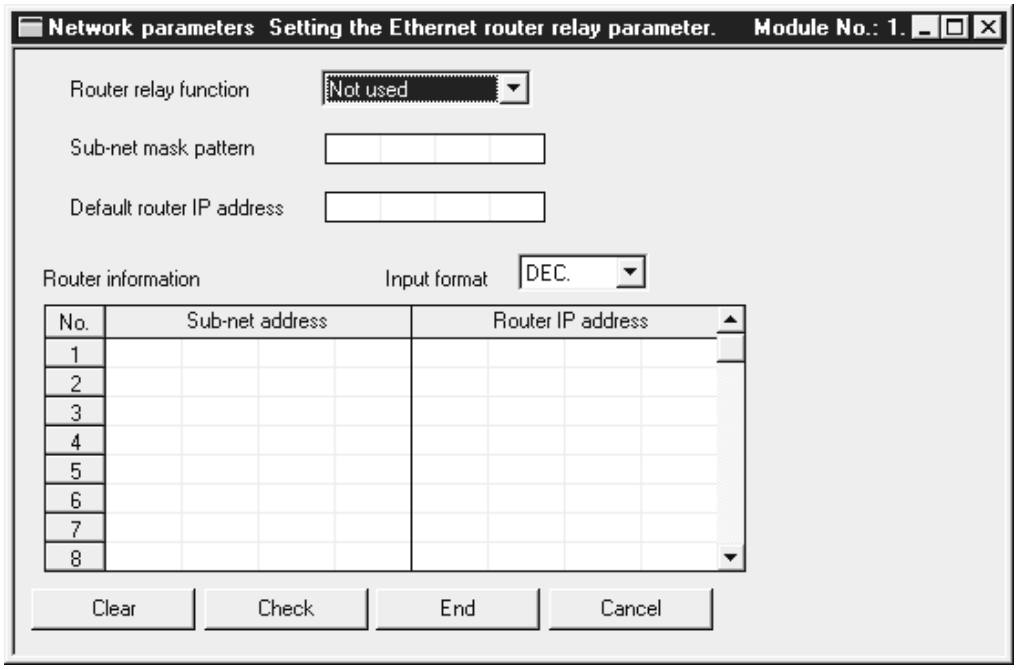
(Sample Program)
The following figure shows a sample program that performs a re-initial processing by writing directly to buffer memory.
Create a program that contains contacts (user flags, etc.) indicating the conditions shown below within a sequence program.
• A contact where the initial processing completion signal (X19) is ON
• Open completion signal storage area (address: 5000H (20480))



POINT
This is an example program that can be used to perform re-initial processing when communicating with connections 1 and 2. The corresponding numbers and bits for each connection should be specified if other connections are used.

5.3 Router Relay Parameter

This section explains the router relay parameter using GX Developer.
Select [Setting the number of MNET/10H Ethernet cards] – [Router relay parameter] to start the [Setting the Ethernet router relay parameter] screen.



Item name	Description of setting	Setting range/options
Router relay function	Select whether the router relay function is not used or is used.	• Use • Not used
Sub-net mask pattern	Set the sub-net mask.	C0000000H to FFFFFFFCH
Default router IP address	Set the IP address of the default router to be routed through.	Other than 00000000H and FFFFFFFFH
Input format	Select the input format of setting items.	• Decimal • Hexadecimal
Sub-net address	Set the sub-net address or network address of the external device when communicating with an external device on another Ethernet via a router other than the default router.	Other than 00000000H and FFFFFFFFH
Router IP address	Set the IP address of a router used when communicating with an external device on another Ethernet via a router other than the default router.	—

(1) Router relay function (Address: 4H ... b5, b4)

(a) Set whether the router relay function will be used or not.
The router relay function is not needed when the QJ71MT91 communicates with the target device on the same Ethernet (the subnet address of the IP address is the same).

(b) The router relay function allows communication with devices on other Ethernets via routers and gateways.
(The router relay function does not mean a function with which the QJ71MT91 acts as a router.)

(c) One default router and a maximum of any eight routers can be set for the router relay function.

(2) Subnet mask pattern (Address: 200H, 201H)

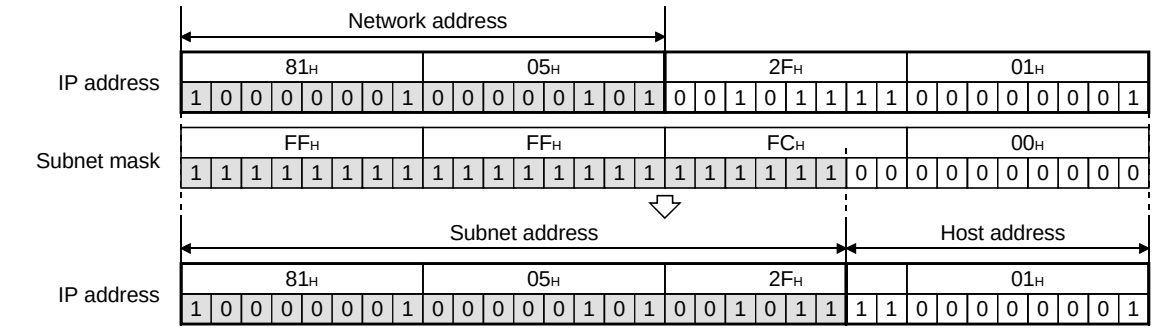
(a) Set the subnet mask. ∗1 (Setting range: C0000000H to FFFFFFFCH)
Consult the network administrator for the setting.

(b) When not using the subnet mask, set any of the following table values according to the class.

Class	Mask value
Class A	FF000000H
Class B	FFFF0000H
Class C	FFFFFF00H

∗1 Networks constructed by Ethernet include small-scaled network systems where multiple devices are connected to one Ethernet, and medium- and large-scaled network systems where multiple small-scaled networks are connected by routers, etc.
The subnet mask logically divides one network, where many devices are connected, into multiple sub-networks to facilitate administration.

(Example) Class B



POINT

(1) All devices on the same sub-network must have common subnet masks.
(2) When not administrated by the sub-network, the connected devices need not have subnet masks. (Set the network address of the corresponding class.)

(3) Default router IP address (Address: 202H, 203H)

Set the IP address of the router (default router) to be used when the QJ71MT91 communicates with the target device on another Ethernet via other than the router specified in the router information (refer to (4) below).

Set the value that satisfies the following conditions.

- Condition 1: The IP address class is any of A, B and C.
- Condition 2: The sub-net address of the default router is the same as that of the local station Ethernet module.
- Condition 3: The host address bits are not all "0" or all "1".

POINT

If the corresponding sub-net address does not exist in the router information (refer to (4) in this section) when the connection is opened or data communication is made, communication is made via the default router.

(4) Router information: Subnet address (Address: 205H, 206H and later)

(a) Set the network address (∗1) or subnet address (∗2) of the target device when the QJ71MT91 communicates with the target device on another Ethernet via other than the default router.

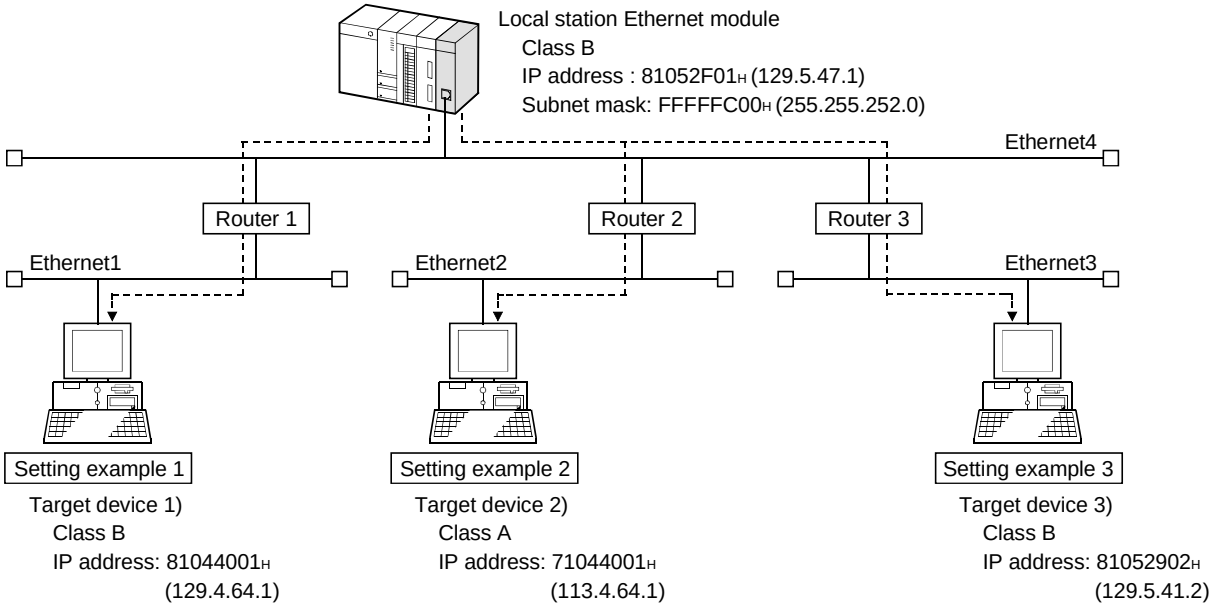
Set the value that satisfies the following conditions.

- Condition 1: The IP address class is any of A, B and C.
- Condition 2: The host address bits are all "0".

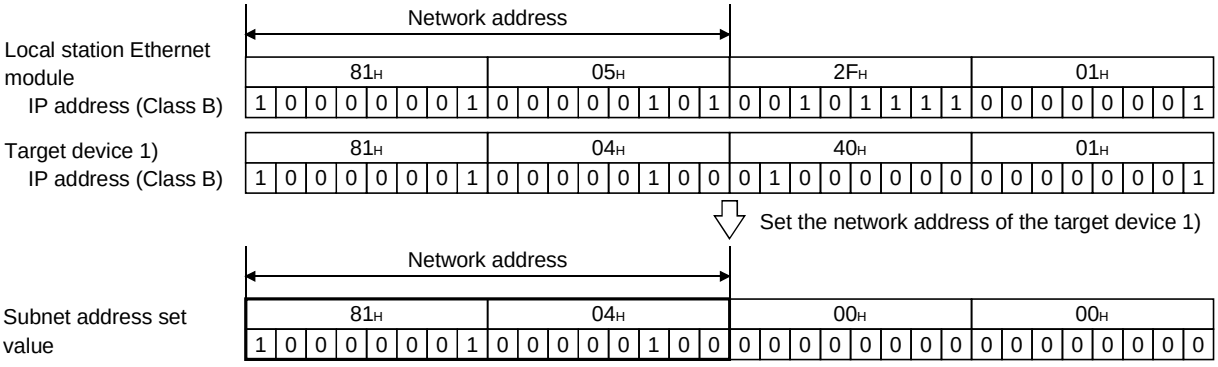
∗1 If the class (network address) of the local station differs from that of the external device, set the network address of the external device.

∗2 If the class (network address) of the local station is the same as that of the external device, set the sub-net address of the external device.

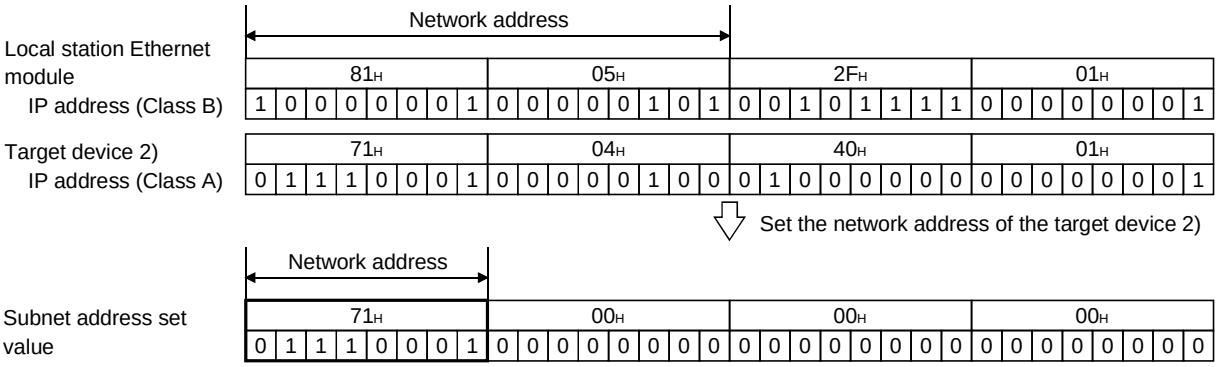
(b) Subnet address setting examples



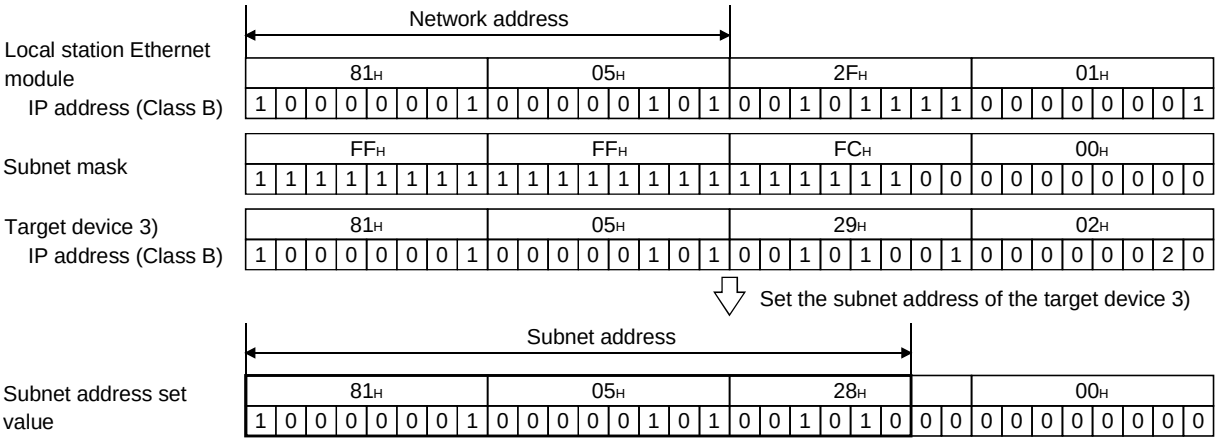
(Setting example 1) When the network addresses of the local station Ethernet module and target device differ


Network address
Local station Ethernet module
IP address (Class B)
81H 05H 2FH 01H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 1
Target device 1)
IP address (Class B)
81H 04H 40H 01H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
Set the network address of the target device 1)
Network address
Subnet address set value
81H 04H 00H 00H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(Setting example 2) When the classes of the local station Ethernet module and target device differ


Network address
Local station Ethernet module
IP address (Class B)
81H 05H 2FH 01H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 1
Target device 2)
IP address (Class A)
71H 04H 40H 01H
0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
Set the network address of the target device 2)
Network address
Subnet address set value
71H 00H 00H 00H
0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(Setting example 3) When the network addresses of the local station Ethernet module and target device are the same


Network address
Local station Ethernet module
IP address (Class B)
81H 05H 2FH 01H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 1
Subnet mask
FFH FFH FCH 00H
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
Target device 3)
IP address (Class B)
81H 05H 29H 02H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 2 0
Set the subnet address of the target device 3)
Subnet address
Subnet address set value
81H 05H 28H 00H
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0

(5) Router information: Router IP address (Address: 207H, 208H and later)

Set the IP addresses of the routers to be used when the QJ71MT91 communicates with the target devices on other Ethernets via other than the default router.

Set the value that satisfies the following conditions.

- Condition 1: The IP address class is any of A, B and C.
- Condition 2: The sub-net address of the router is the same as that of the local station Ethernet module.
- Condition 3: The host address bits are not all "0" or all "1".

POINT
(1) When the Ethernet module communicates with the external device via the router in Passive open (TCP/IP) status, communication can be made without use of the router relay function.
(2) The router relay function is not needed in a system that uses the Proxy router.

5.4 Confirming the Completion of the Initial Processing

The initial processing of the Ethernet module completes by saving the following parameters in the PLC CPU of the station to which the Ethernet module is installed and by restarting the PLC CPU.
(When the processing is completed normally, the [INIT.] LED on the front of the Ethernet module turns on.)

- "Network parameters Setting the number of MNET/10H Ethernet cards" parameter
- "Operational settings" parameter
- "Initial settings" parameter

This section explains how to check the completion of the initial processing.

POINT
The status of the Ethernet module becomes communication enabled when the initial processing is completed normally. See reference sections for each communication function to perform communication. When the initial processing has not been completed normally, do the following to check the contents of the error, take corrective actions, then execute the initial processing again. • Check the error code using the "Parameter status" of the Ethernet diagnostics. (See Section 11.2.) • Check the contents of the error corresponding to the error code, then take corrective actions. (See Section 11.3.3.)

5.4.1 PING test using GX Developer (Via Ethernet board)

This section explains how to check the completion status of the initial processing for the Ethernet module using the PING test function of the GX Developer Ethernet diagnostics.

(1) PING test

(a) The PING test is used to check, through GX Developer, the existence of an Ethernet module on the same Ethernet line that has completed the initial processing (*1) or an external device (such as a personal computer) with the designated IP address.

*1 The PING test can also be performed for the QnA/A Series Ethernet modules. However, the PING test can be performed for the following A Series Ethernet modules starting with software version S: AJ71E71, AJ71E71-B2, AJ71E71-B5

(b) The following can be checked by performing the PING test for the Ethernet module:

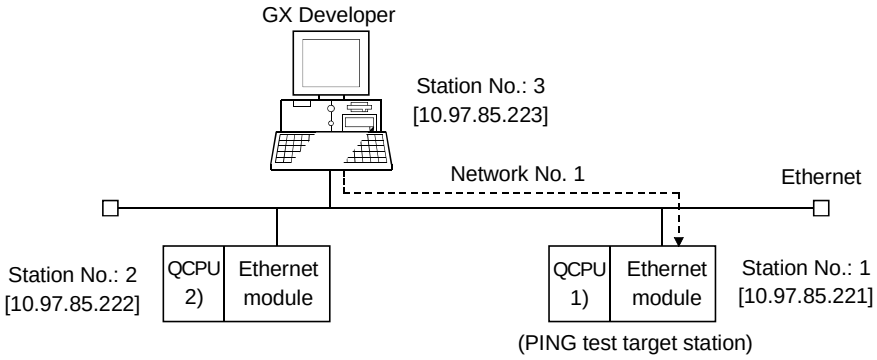
- Whether a line has been properly connected to the test target Ethernet module
- Whether the parameters for the Ethernet module have been correctly set
- Whether the initial processing for the Ethernet module has been completed normally

(c) The PING test can be performed for an Ethernet module in the same Ethernet as the local station (same sub-net address.)

(d) If the external device subject to the PING test is an Ethernet module, do not designate the UDP port for GX Developer of the Ethernet module as a valid port for the remote password check. If it is designated as a valid port for the remote password check, the PING test cannot be performed.

(2) Executing the PING test

The example below explains the PING test methodology and GX Developer settings when performing a PING test for the Ethernet module in the same Ethernet from a peripheral device.



(a) Setting the PING test target station

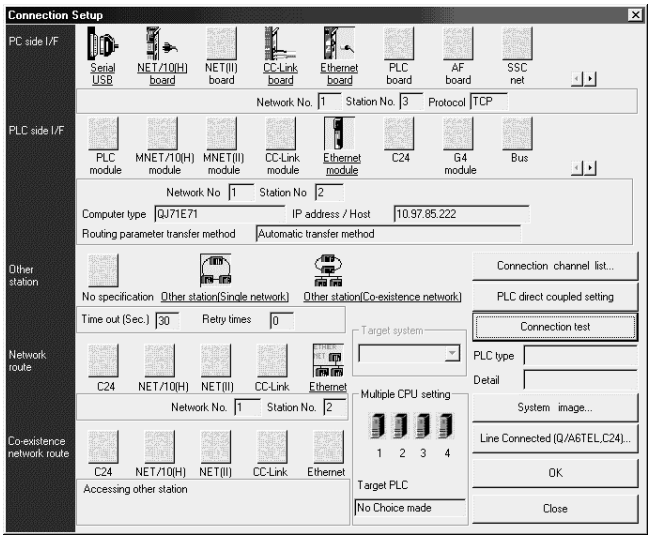
1) Set the following Ethernet module parameters for the PING test target station through GX Developer.
Use default values for setting items other than those listed below.

Setting screen	Setting item	Setting description	
		QCPU 1)	QCPU 2)
Network parameters setting the number of MNET/10H Ethernet cards	Network type	Ethernet	Ethernet
	Starting I/O No.	0000	0000
	Network No.	1	1
	Group No.	1	1
	Station No.	1	2
Operational setting	IP address	[10.97.85.221]	[10.97.85.222]

2) Write the parameters to the applicable station.

3) The initial processing is completed when the PLC CPU restarts.
(When the initial processing is completed normally, the [INIT.] LED of the Ethernet module lights up.)

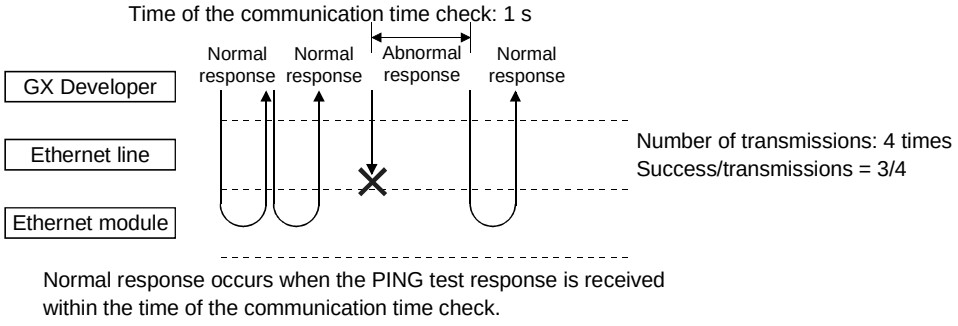
(b) Designating GX Developer connection destination (connecting to QCPU [2])



(c) Executing the PING test through GX Developer

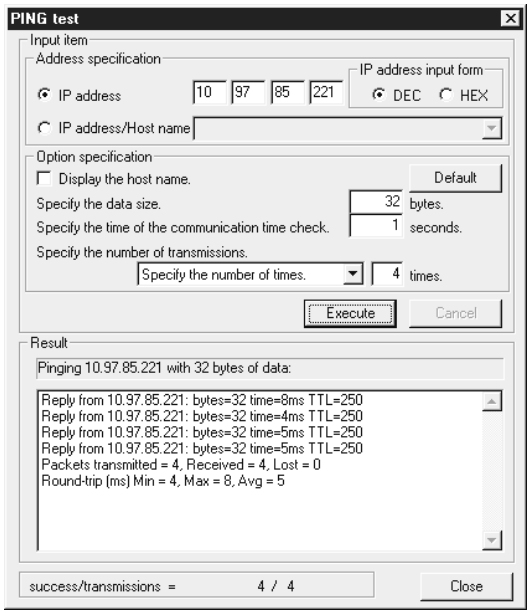
1) Select the PING test on the Ethernet diagnostics screen.
GX Developer → [Diagnostics] → [Ethernet diagnostics] → PING test

2) Perform the settings indicated below, then click the Execute button.
The execution results of the PING test are displayed.
(Example) The following shows the flow of the PING test when "4" is designated as the transmission count.

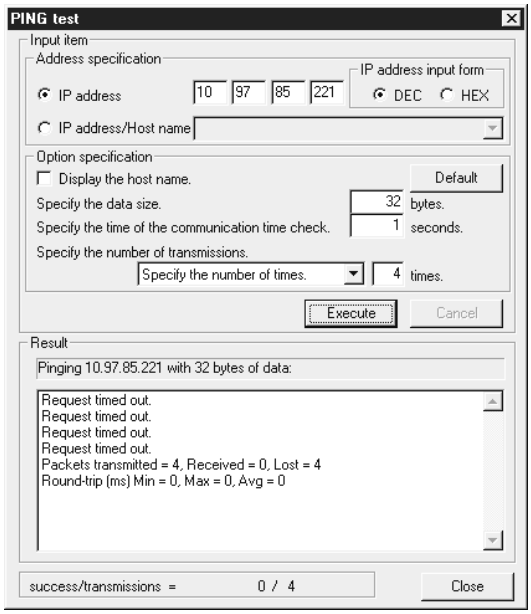


Normal response occurs when the PING test response is received within the time of the communication time check.

[PING test screen]



(Example of normal completion)



(Example of abnormal completion)

[Display contents]

Item name		Description of item setting	Setting range/options
Address specification	IP address	Specify the IP address for the PING test target station.	(Target station IP address)
	IP address input form	Select the input format for the IP address.	Decimal/hexadecimal
	Host name	Specify the host name for the PING test target station.	—
Option specification	Display the host name	Results are displayed using the host name corresponding to the IP address in the result display field.	—
	Specify the data size	Specify the size of the system data transmitted during the PING test.	1 to 8192 bytes (Specify 1460 bytes or less for the Ethernet module.)
	Specify the time of the communication time check	Specify the completion wait time for the PING test.	1 to 30 s
	Specify the number of transmission	Specify the transmission count.	• Specify the number of times. • Execute till interrupting.
Result		Display results of the PING test.	—
Success/transmissions		Display the total packet transmission count and the number of successes during the PING test.	—

(Address specification)
The PING test target station (external device subject to the PING test) is specified by the IP address or the host name.

1) Specification using the IP address
   - Select the input format for the IP address (select: Decimal or hexadecimal)
   - Specify the IP address of the external device according to the input format (decimal or hexadecimal).

2) Specification using the host name
   Specify the host name of the external device set in the DNS server or the HOSTS file for the personal computer on which GX Developer is mounted.
   ✻ The IP address can also be entered in the host name specification field.

(Option specification)
Set the details for the PING test. (No setting required if the default is used.)

1) Display the host name.
   Select this to display the host name instead of the IP address for the PING test destination device in the result display field.

2) Specify the data size.
   Specify the size of the system data to be transmitted for the test during the PING test.
   Input range: 1 to 8192 bytes (default: 32 bytes)
   ✻ The Ethernet module will return a response of 1460 bytes if the PING test is performed when a data size of 1460 bytes or greater for transmitting to the Ethernet module is specified.

3) Specify the time of the communication time check.
   Specify the response wait time for the PING test.
   Input range: 1 to 30 s (default: 1 s)

4) Specify the number of transmissions.
   Specify the number of times the PING test is to be performed.

Selection item	Description of item	Remarks
Specify the number of times	The PING test is performed for the number of specified times.	Transmission count: 1 to 50 times (default: 4 times)
Execute till interrupting	The PING test is performed until the interrupt button is pressed.	—

(Result)
Results of the PING test are displayed.

<When the test is completed abnormally>
Check the following, then perform the PING test again.
- How the Ethernet module is mounted on the base unit.
- Status of the connection to the Ethernet.
- Contents of the parameters written to the PLC CPU.
- Operating status of the PLC CPU (whether any errors have occurred).
- IP addresses set in GX Developer and the PING test target station.
- Whether the external device has reset when the Ethernet module was changed.

(Success/transmissions)
The number of successes and the total packet transmission count when the PING test is performed are displayed.

5.4.2 PING test using GX Developer (Via CPU)

This section explains how to check the completion status of the initial processing for the Ethernet module using the PING test function of GX Developer's Ethernet diagnostics.

(1) PING test

(a) The PING test is used to check the existence of an Ethernet module on the same Ethernet line as the connection station of GX Developer that has completed the initial processing (*1) or an external device (such as a personal computer) with the designated IP address.

*1 The PING test can also be performed for the QnA/A Series Ethernet modules. However, the PING test can be performed for the following A Series Ethernet modules starting with software version S:
AJ71E71, AJ71E71-B2, AJ71E71-B5

(b) The following items can be checked by performing the PING test for the Ethernet module:

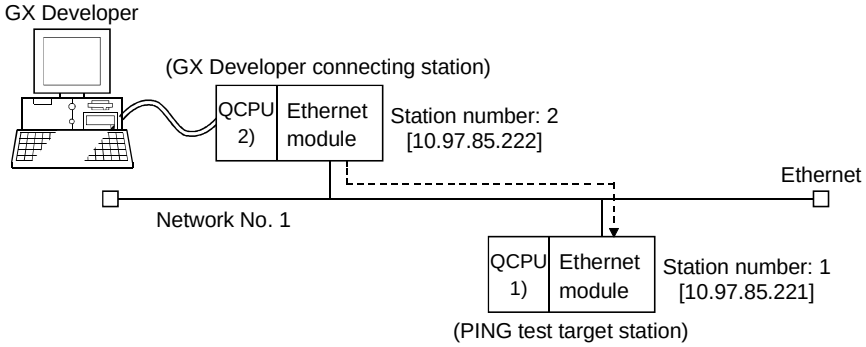
- Whether a line has been properly connected to the test target Ethernet module.
- Whether the parameters for the Ethernet module have been correctly set.
- Whether the initial processing for the Ethernet module has been completed normally.

(c) The PING test can be performed for an Ethernet module on the same Ethernet as that of the local station (same sub-net address).
Note that the PING test is not available for the Ethernet module mounted on the station to which GX Developer is directly connected (local station). (The PING test addressed to the local station is not available.)

(d) If the external device subject to the PING test is an Ethernet module, do not designate the UDP port for GX Developer of the Ethernet module as a valid port for the remote password check. If the port is designated as such, the PING test cannot be performed.

(2) Executing the PING test

The example below explains the PING test methodology and GX Developer settings when a PING test is performed for an Ethernet module of other station from GX Developer connected to the QCPU.



(a) Settings on the QCPU station side

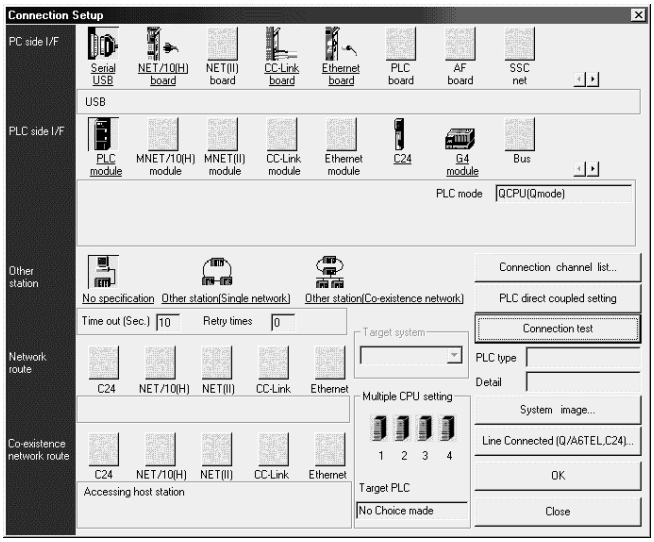
1) Set the following Ethernet module parameters for each QCPU using GX Developer.
Use default values for setting items other than those listed below.

Setting screen	Setting item	Setting description	
		QCPU 1)	QCPU 2)
Network parameters setting the number of MNET/10H Ethernet cards	Network type	Ethernet	Ethernet
	Starting I/O No.	0000	0000
	Network No.	1	1
	Group No.	1	1
	Station No.	1	2
Operational setting	IP address	[10. 97. 85. 221]	[10. 97. 85. 222]

2) Write the parameters to the applicable station.

3) The initial processing is completed when the PLC CPU restarts.
(When the initial processing is completed normally, the [INIT.] LED of the Ethernet module lights up.)

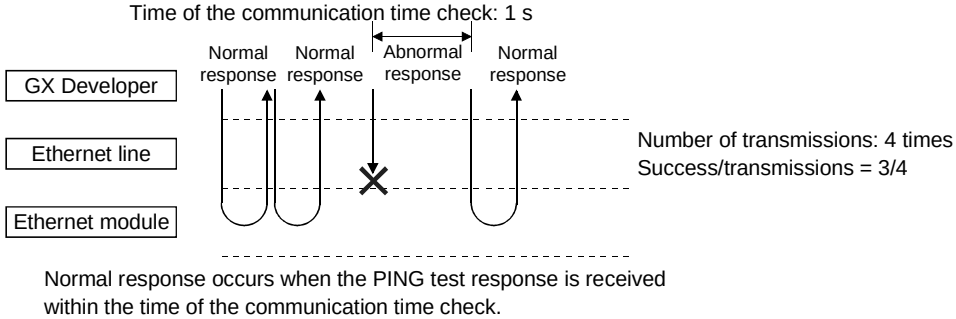
(b) Designating the GX Developer connection destination (connecting to QCPU 2)



(c) Executing the PING test through GX Developer

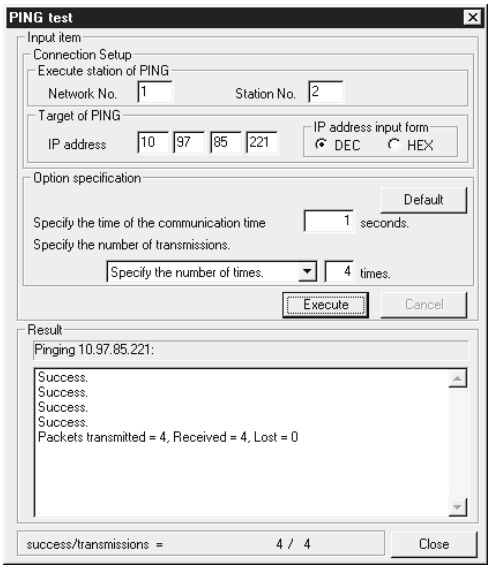
1) Select the PING test on the Ethernet diagnostics screen.
GX Developer → [Diagnostics] → [Ethernet diagnostics] → PING test

2) Perform the settings indicated below, then click the Execute button.
The execution results of the PING test are displayed.
(Example) The following shows the flow of the PING test when "4" is designated as the transmission count.

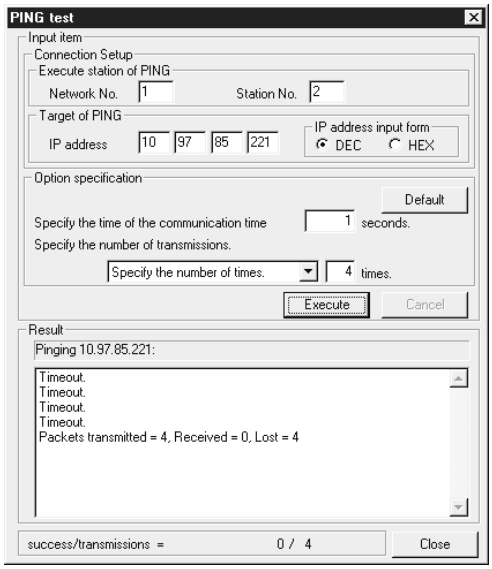


Normal response occurs when the PING test response is received within the time of the communication time check.

[PING test screen (via CPU)]



(Example of normal completion)



(Example of abnormal completion)

[Display contents]

Item name		Description of item setting	Setting range/option
Execute station of PING	Network No.	Specify the network number of the Ethernet module in the PING executing station.	1 to 239
	Station No.	Specify the station number of the Ethernet module in the PING executing station.	1 to 64
Target of PING	IP address	Specify the IP address of the PING test target station.	00000001_H to $FFFFFFFE_H$
	IP address input form	Select the input format of the IP address.	Decimal/hexadecimal
Option specification	Specify the time of the communication time	Specify the response wait time for the PING test.	1 to 30 s
	Specify the number of transmissions	Specify the transmission count.	• Specify the number of times. • Execute till interrupting.
Result		Display the result of the PING test.	—
success/transmissions		Display the total packet transmission count and its success count during the PING test execution.	—

(Connection Setup)

1) Execute station of PING (station connected to GX Developer)
   Specify the network number and station number of the Ethernet module that will execute the PING test.
   The network number and the station number will be set on the " network parameters setting the number of MNET/10H Ethernet cards" screen of GX Developer.

2) Target of PING
   Specify the IP address of the PING test target station (the External device subject to the PING test).
   - Select the input format of the IP address (select decimal or hexadecimal).
   - Specify the IP address of the PING test target station according to the input format (decimal or hexadecimal).

(Option specification, result, success/transmissions)
The information displayed is the same as that displayed when performing a PING test via the Ethernet board. See Section 5.4.1.

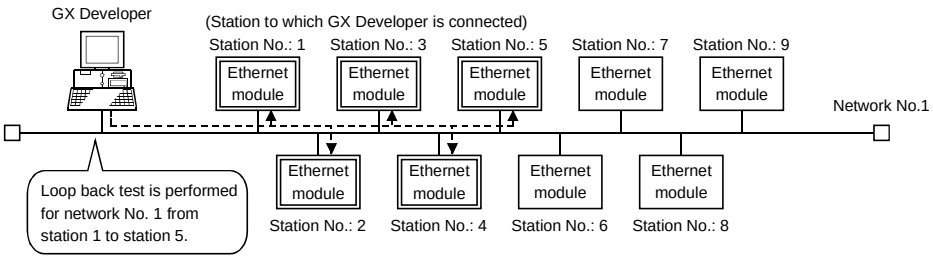
5.4.3 Loop back test using GX Developer

This section explains how to check the completion status of the initial processing for the Ethernet module using the loop back test function of GX Developer Ethernet diagnostics.

(1) Loop back test

(a) The loop back test is performed in the network for GX Developer connection destination. The test involves transmission of loop back test messages in order with respect to the network specified by the loop back test request destination and all Ethernet modules within the station number range (function version B and later) to check whether the initial processing for each module has been completed.

* A loop back test can be performed on a network connected via the Ethernet board.
* Since the Ethernet module of function version A and QnA/A series Ethernet module do not have a function to respond to this request, the test results cannot be checked.



(b) The following can be checked by performing the loop back test:

- Whether a line has been properly connected to the test target Ethernet module
- Whether the parameters for the Ethernet module have been correctly set
- Whether the initial processing for the Ethernet module has been completed normally

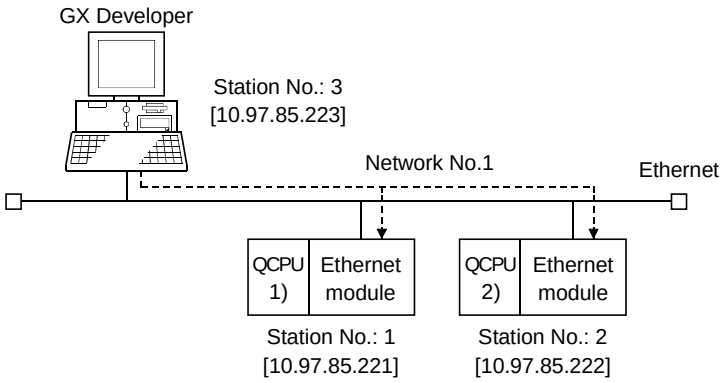
(c) The loop back test can be performed for an Ethernet module in the same Ethernet as the local station (same sub-net address).

(d) In the station on which the Ethernet module subject to the loop back test is mounted, do not designate the UDP port for GX Developer of the Ethernet module as a valid port for the remote password check. If it is designated as a valid port for the remote password check, the loop back test cannot be performed.

(e) When performing a loop back test using GX Developer, set the router relay function in the router relay parameter of GX Developer to "Not used."

(2) Executing the loop back test

The example explains the loop back test methodology and GX Developer settings when performing a loop back test for the Ethernet module in the same Ethernet from the GX Developer.



(a) QCPU station side settings

1) Set the following Ethernet module parameters for each PLC CPU through GX Developer.
Use default values for setting items other than those listed below.

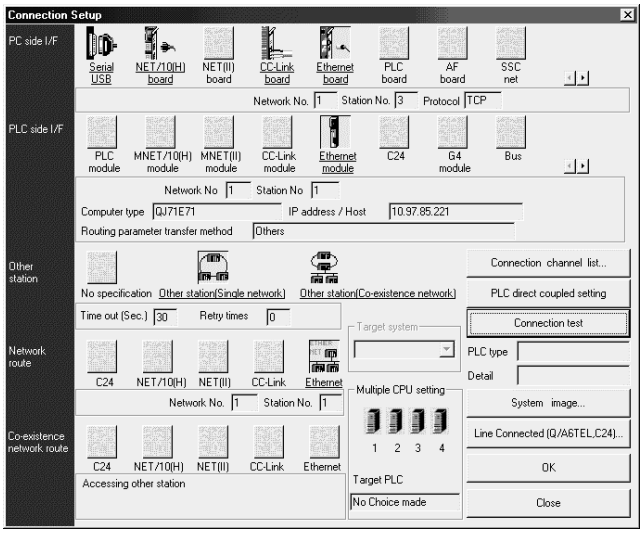
Setting screen	Setting item	Setting description		Remarks
		QCPU 1)	QCPU 2)	
Network parameters setting the number of MNET/10H Ethernet cards	Network type	Ethernet	Ethernet	See Section 4.6 Set the local station network No., etc.
	Starting I/O No.	0000	0000	
	Network No.	1	1	
	Group No.	1	1	
	Station No.	1	2	
Operational setting	IP address	[10.97.85.221]	[10.97.85.222]	See Section 4.7 Enter the local station IP address.
MNET/10 routing information	MNET/10 routing system	Table exchange system	Table exchange system	See Section 3.3.1 of User's Manual (Application).
	Network No.	1	1	
	Station No.	3	3	
	IP address	[10.97.85.223]	[10.97.85.223]	

* It is not necessary to set the network No., station No. and IP address for GX Developer (station performing the loop back test) when "Automatic response system" is specified as the MNET/10 routing system under the MNET/10 routing information.

2) Write the parameters to the applicable station.

3) The initial processing is completed when the PLC CPU reboots.
(When the initial processing is completed normally, the [INIT.] LED of the Ethernet module lights up.)

(b) Designating GX Developer connection destination (connecting to QCPU [1])

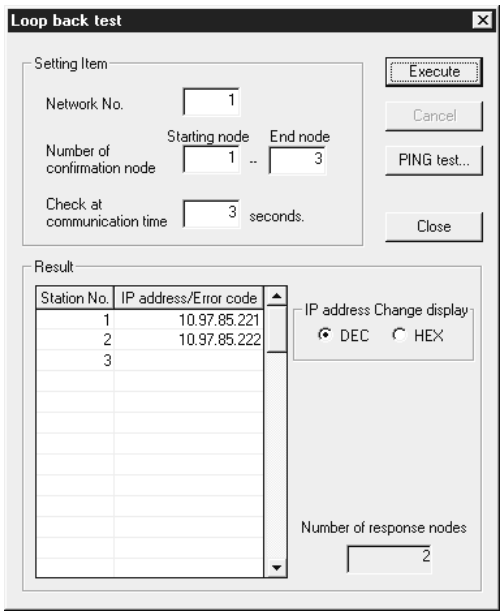


(c) Executing the loop back test through GX Developer

1) Select the loop back test on the Ethernet diagnostics screen.
GX Developer → [Diagnostics] → [Ethernet diagnostics] → Loop back test

2) Perform the settings indicated below, then click the Execute button.
The execution results of the loop back test are displayed.

[Loop back test screen]



[Display contents]

Item name		Description of item setting	Setting range/options
Setting item	Network No.	Specify the network No. for the loop back test target module.	1 to 239
	Number of confirmation node	Specify the station number range for the loop back test target module.	1 to 64
	Check at communication time	Specify the response wait time for the loop back test.	1 to 99 s

(Network No. (1 to 239))
Specify the loop back test request destination network No.
Input range: 1 to 239

(Number of confirmation node (1 to 64))

1) Specify the range (starting node and end node) of the Ethernet module subject to the loop back test using the station No.
   Input range: 1 to 64
2) The loop back test is performed in order from the starting node.
   ✻ Upon the reception of a response from the station performing the loop back test or after the time of the communication time check elapses, the loop back test is performed for the next target station and so on in order until the end station.

(Check at communication time)
Specify the response wait time for the loop back test for each station.
Input range: 1 to 99 s (default: 10 s)

(Result)
Results of the loop back test are displayed.

1) The IP address for the Ethernet module that has completed the loop back test normally is displayed.
   ✻ If the same IP address or station number has been set to multiple stations due to a parameter setting error on the loop back test target module side, only the results for the station that issued a response first will be displayed.
2) "No response" or an error code will be displayed for an Ethernet module in which the loop back test is completed abnormally.
   ✻ "No response" is not shown with GX Developer Version 6.03D or former products.
3) If the local station is specified as a loop back test target station (station performing a loop back test), "No response" will be displayed as the loop back test results for the local station.
   ✻ "No response" is not shown with GX Developer Version 6.03D or former products.

(d) Corrective action when the test is completed abnormally
The table below shows the display contents, status of the target Ethernet module, causes and corrective actions when the loop back test has been completed abnormally.

Display contents of loop back test results	Status of the target Ethernet module	Cause	Corrective action
"IP address"	Initial processing normal completion status ([INIT.] LED ON status)	—	—
"No response"	No error	The initial processing for the target Ethernet module has not been completed normally.	Check the setting values for the following parameters: • Network parameters setting the number of MNET/10H Ethernet cards • Operational setting • Initial setting
		There is an error in the line connection to the target Ethernet module. (Cable disconnection, breakage, etc.)	• Check the cable. • Check the transceiver.

Display contents of loop back test results	Status of the target Ethernet module	Cause	Corrective action
"No response"	No error	IP address for the target Ethernet module is incorrect. (Class or sub-net address differs from the Ethernet module setting.)	Check the parameter setting values for the operation setting.
		The same IP address has been set to multiple target Ethernet modules.	
		The same network number or station number has been set to multiple target Ethernet modules.	Perform a PING test for the "No response" module. If the test has been completed normally, check the parameter setting value for the network parameters setting the number of MNET/10H Ethernet cards.
	Error exists	The Ethernet line is in the high load status (including a case in which an error equivalent to the error code C030H, C031H is occurring).	Perform another test when the load on the Ethernet line is low.
		Routing parameter is not set. (An error equivalent to the error code C080H is occurring.)	Check the setting value for the routing parameter. (See Chapter 3 of User's Manual (Application)
"Error code"	No error	The remote password status of GX Developer UDP for the target Ethernet module is in the lock status.	Cancel the remote password setting and write the parameter to the PLC CPU.
		The target Ethernet module is a function version A module.	Check the model and function version for the subject module.
	Error exists	The Ethernet line is in the high load status (including a case in which an error equivalent to the error code C030H, C031H is occurring)	Perform another test when the load on the Ethernet line is low.

POINT

(1) The loop back test can only be performed for the Ethernet module of function version B and later.
Test results cannot be checked for the following modules. ("No response" will be displayed. However, nothing is shown under the use of GX Developer version 6.03D or former products.)
Also, they will not be counted in the total number of stations.
- Ethernet module of function version A, QnA/A series Ethernet module (This is because there is no function to respond to this request.)
- Ethernet module that has not completed the initial processing (This is because the initial processing has not been completed.)

＊ If the Ethernet module of function version A or QnA/A series Ethernet module that does not have a function to respond to the loop back test receives a loop back test request, the following error codes will be stored in the error log area of the buffer memory.
- Error code 4080H in the case of an Ethernet module of function version A or QnA series Ethernet module of function version B
- Error code 50H in the case of a QnA series Ethernet module of function version A or A series Ethernet module.

(2) After checking the contents of the error and taking corrective actions for an Ethernet module in which the loop back test has been completed abnormally, reboot the station on which the Ethernet module is mounted.
With rebooting, the initial processing for the Ethernet module will be performed.
Check the completion of the initial processing for the Ethernet module using the PING test.
The PING test can also be performed on the "Loop back test" screen.

(3) If an error code is displayed due to abnormal completion of the loop back test, check the contents of the error corresponding to the error code, then take corrective actions by referring to Section 11.3.3.

5.4.4 PING command (Personal computer → Ethernet module)

The following example illustrates how to confirm the completion of the initial processing by issuing the PING command to the local station's Ethernet module from an external device connected on the same Ethernet. (In the example, the confirmation is made between devices whose IP address class and sub-net address are identical.)

<Designation method>

ping IP address

<Example>

IP address of the Ethernet module: 192.0.1.254

Example of screen at normal completion

```
C:\>ping 192.0.1.254…Execute the ping command

Pinging 192.0.1.254 with 32 bytes of data:

Reply from 192.0.1.254: bytes=32 time=1ms TTL=128
Reply from 192.0.1.254: bytes=32 time<10ms TTL=128
Reply from 192.0.1.254: bytes=32 time<10ms TTL=128
Reply from 192.0.1.254: bytes=32 time<10ms TTL=128

Ping statistics for 192.0.1.254:
        Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>_
```

Example of screen at abnormal completion

```
C:\>ping 192.0.1.254…Execute the ping command

Pinging 192.0.1.254 with 32 bytes of data:

Request timed out:
Request timed out:
Request timed out:
Request timed out:

Ping statistics for 192.0.1.254:
        Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>_
```

<When the PING command does not respond successfully>

Check the following items and send the PING command again.

- Check the Ethernet module's attachment to the base unit.
- Check the connection to the Ethernet network.
- Check the contents of each parameter written to the PLC CPU.
- Check the operation condition of the PLC CPU (are there any irregularities?).
- Check the IP address of the Ethernet module dictated by the PING command.

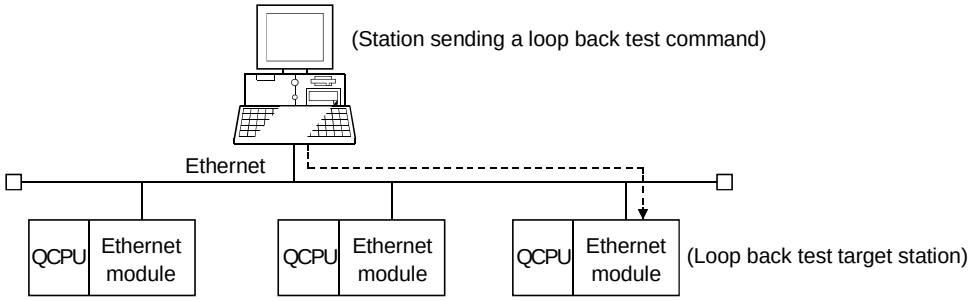
5.4.5 Loop back test (Communication using the MC protocol)

The loop back test can be performed with communication using the MC protocol in order to check the completion status of the initial processing for the target Ethernet module.
The following is an overview of the loop back test for communication using the MC protocol. See the Q Corresponding MELSEC Communication Protocol Reference Manual for details.

(1) Loop back test for communication using the MC protocol
The loop back test is a function designed to check the following:
- Whether a line has been properly connected to the test target Ethernet module.
- Whether the parameters for the Ethernet module have been correctly set.
- Whether the initial processing for the Ethernet module has been completed normally.
- Whether the program for the external device is running correctly.

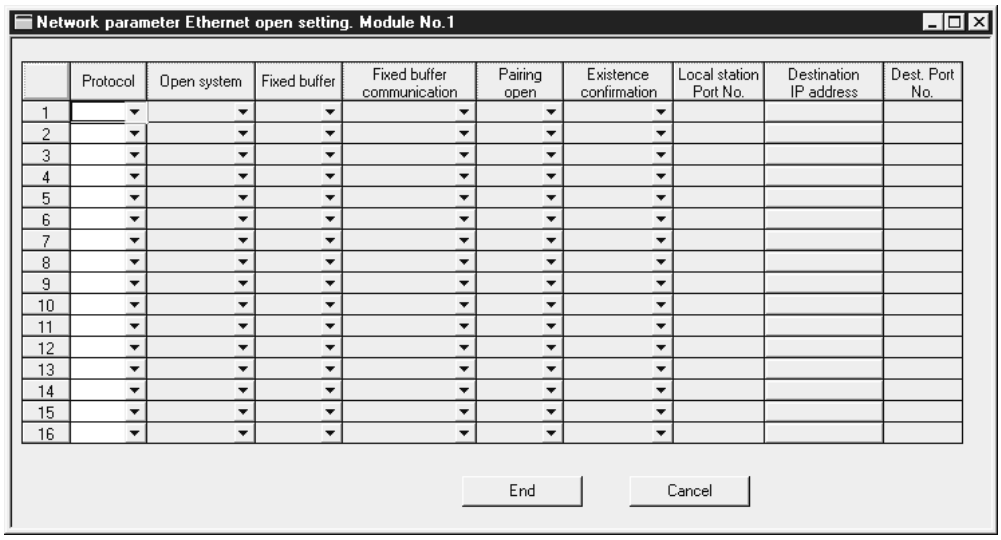
(2) It is necessary to connect lines when performing communication using the MC protocol with the user port on the Ethernet module side.
Perform the open processing for the connection to be used on the Ethernet module side.

(3) This function can only be used for the Ethernet module of the local station. The function cannot be used for the Ethernet module of another station via a network system.



5.5 Open Settings

This section explains the open setting using GX Developer.
Select [Setting the number of MNET/10H Ethernet cards] – [Open settings] to start the [Ethernet open settings] screen.



A sequence program can perform open processing (establishing connection) with external devices up to a maximum of 16 stations
Once a connection is established with an external device, it is possible to communicate using the MC protocol, fixed buffer communication, and random access buffer communication.
Thus, the open processing is required even when communication using the MC protocol and random access buffer communication.

Item name	Description of setting	Setting range/options
Protocol	Set the communication method (protocol).	• TCP/IP • UDP/IP
Open system	Select the connection open system.	• Active open • Unpassive open • Fullpassive open • MELSOFT connection • OPS connection
Fixed buffer	Select the usage of the fixed buffer.	• Send • Receive
Fixed buffer communication	Select which protocol is used for fixed buffer communication.	• Procedure exist • No procedure
Pairing open	Select whether paring open is used or not.	• Pairs • No pairs
Existence confirmation	Select whether the continued existence of a destination station for a connection should be confirmed or not.	• No confirm • Confirm
Local station Port No.	Set the local station's port No.	401_H to 1387_H or $138B_H$ to $FFFE_H$
Destination IP address	Set the IP address of an external device.	1_H to $FFFFFFFF_H$ ($FFFFFFFF_H$: Simultaneous Broadcast)
Dest. Port No.	Set the port No. of an external device.	401_H to $FFFF_H$ ($FFFF_H$: Simultaneous Broadcast, can be set only when receiving)

(1) Protocol
(connection numbers 1 to 8; addresses: 20H to 27H … b8)
(connection numbers 9 to 16; addresses: System area is used)

(a) Select the protocol for each connection.

Name of setting	Description of setting
TCP	Communicate using TCP/IP.
UDP	Communicate using UDP/IP.

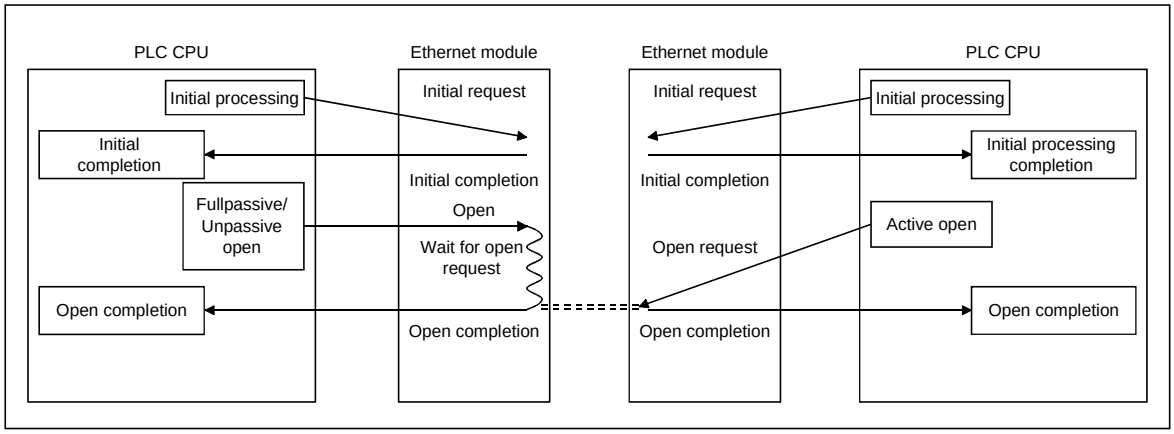
(b) For protocols (TCP/UDP), see 1.4, "Software Configuration".

(2) Open system
(connection numbers 1 to 8; addresses: 20H to 27H … b15, b14)
(connection numbers 9 to 16; addresses: System area is used)

(a) Select the connection open system for each connection for which "TCP" is selected in "(1) Protocol". If "UDP" is selected, the specification of this item is not required.

Name of setting	Description of setting
Active	Perform active open processing to an external device that waits for a passive open (Fullpassive/Unpassive) on the TCP connection.
Unpassive	Perform passive open processing on the TCP connection addressing all the devices connected to a network. (The local station is placed in the wait status to wait for an Active open request to be sent.
Fullpassive	Perform passive open processing on the TCP connection, only addressing specific devices. (The local station is placed in the wait status to wait for an Active open request to be sent.) The local station waits for an Active open request from the opposite station set in "(8) Destination IP address".
MELSOFT connection ∗1 ∗2 ∗3	Used to connect MELSOFT products via TCP/IP communication. Perform passive open processing on the TCP connection, addressing all the MELSOFT products connected to a network. (The local station is placed in the wait status for an Active open request to be sent.)
OPS connection ∗1 ∗4	Used to connect the OPS via TCP/IP communication. Perform passive open processing on the TCP connection, addressing only specific OPSs. (The local station waits for an Active open request from the OPS set in "(8) Destination IP address".)

∗1 Regardless of the initial timing setting in the operation setting (refer to Section 4.7), this connection will always wait for the open status.
∗2 The set connection is dedicated to data communication with the MELSOFT products.
∗3 When simultaneously connecting to multiple MELSOFT products, set the connections as many as the number of MELSOFT products. (Setting is unnecessary when only one product is connected. The connection dedicated to the system is used.)
∗4 When using the MELSOFT product, such as GX Developer, on the OPS to make TCP/IP communication with the Ethernet module, use the dedicated connection for system (GX Developer communication TCP port) or set "MELSOFT connection" in this setting.



(3) Fixed buffer
(connection numbers 1 to 8; addresses: 20H to 27H … b0)
(connection numbers 9 to 16; addresses: System area is used)

(a) Here it is selected whether the fixed buffer corresponding to each applicable connection number will be used for sending or receiving when communicating using the fixed buffers.

Name of setting	Description of setting
Send	For sending or fixed buffer communication is not executed.
Receive	For receiving.

(b) When both sending and receiving are performed with an external device using fixed buffer communication, one buffer for sending and one for receiving are required. Thus, two connections should be set.

(c) Whether the fixed buffers are set for sending or receiving, external devices can communicate using the MC protocol or the random access buffers.

(4) Fixed buffer communication
(connection numbers 1 to 8; addresses: 20H to 27H … b9)
(connection numbers 9 to 16; addresses: System area is used)

(a) For this item, select the communication method when communicating using the fixed buffers.

Name of setting	Description of setting
Procedure exist	• In fixed buffer communication, data is communicated in 1:1 by handshaking with the external device. • Communication using the MC protocol and the random access buffers can be performed as well.
No procedure	• The no procedure fixed buffer communication uses dedicated connections. • The PLC CPU and external devices communicate data in 1:1 or 1:n mode through simultaneous broadcasting. (* 1) • The handshaking with an external device must be performed using a sequence program.

* 1 For details on the simultaneous broadcasting, see Section 8.3, "Simultaneous Broadcasting When UDP/IP is Used".

(5) Pairing open
(connection numbers 1 to 8; addresses: 20H to 27H … b7)
(connection numbers 9 to 16; addresses: System area is used)

(a) Select whether or not the Ethernet module's receiving and sending connections should be made into one pair and connected to one port of an external device when using fixed buffer communication (both of the procedure exists and no procedure can be designated).
For more details on this, see Section 5.7, "Pairing Open".

Name of setting	Description of setting
No pairs	Does not use the pairing open method.
Pairs	Uses the pairing open method.

(6) Existence confirmation
(connection numbers 1 to 8; addresses: 20H to 27H … b1)
(connection numbers 9 to 16; addresses: System area is used)

(a) This setting selects whether or not the Ethernet module should confirm that an external device still operates normally when there is no communication for a fixed period of time. The open processing for the connection with the external device must have been completed.

Name of setting	Description of setting
No confirm	Do not confirm the existence of the external device.
Confirm	Confirm the existence of the external device. For details on the settings of the existence confirmation time and others, see Section 5.2, "Initial Settings".

(b) When an error occur in the existence confirmation, the Ethernet module performs the following processing.
- Forcefully closes the line and stores the error information in the error log area (addresses: E0H to 1FFH) of the buffer memory.
- Turns off the open completion signal (address: corresponding bit of 5000H) and turns on the open abnormal detection signal (X18).

(c) If the external device will be changed while a UDP/IP connection is open, "No confirm" should be selected.
If "Confirm" is selected, the Ethernet module will confirm the existence of the first destination after the UDP/IP connection is opened. Existence confirmation is not performed for the changed destination, i.e. the newly selected external device.

(d) Select "No confirm", when transmitting data in simultaneous broadcast in the no procedure fixed buffer communication.

(7) Local station Port No.
(connection numbers 1 to 8; addresses: 28H to 5FH)
(connection numbers 9 to 16; addresses: System area is used)

(a) In this item the port number of each connection for the Ethernet module is set in hexadecimal.

(b) The setting values are designated in a range from 401H to 1387H and from 138BH to FFFEH. Set port numbers that are not already used by other port. (Port numbers 1388H to 138AH cannot be designated because they are used by the operating system of the Ethernet module.)

(c) Set the port numbers for the Ethernet module upon consulting a network administrator.

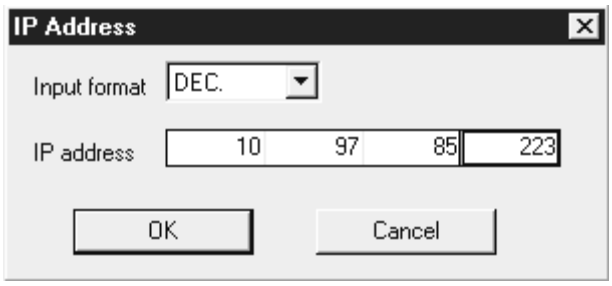
(8) Destination IP address
(connection numbers 1 to 8; addresses: 28H to 5FH)
(connection numbers 9 to 16; addresses: System area is used)

(a) Select the input format of the IP address (options: decimal/hexadecimal).

(b) Set the IP addresses (two words) for external devices in the chosen input format (decimal/hexadecimal).

(c) The IP addresses of external devices must be given values other than 0H. Furthermore, FFFFFFFFH is the setting value for simultaneous broadcast communication.

(d) Set the IP addresses of external devices upon consulting a network administrator.



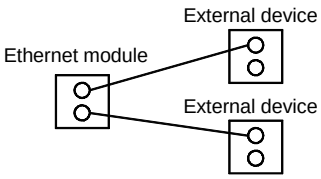
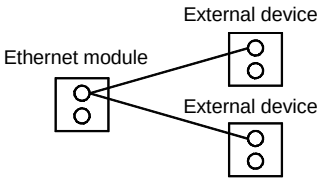
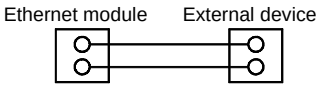
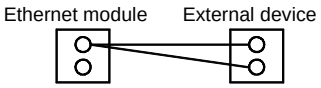
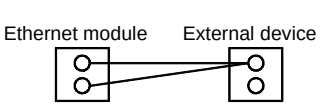
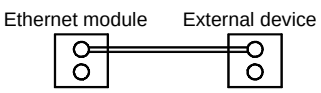
(9) Destination Port No.
(connection numbers 1 to 8; addresses: 28H to 5FH)
(connection numbers 9 to 16; addresses: System area is used)

(a) Set the port numbers of the external devices for each connection in hexadecimal.

(b) The port numbers of the external devices are set in a range from 401H to FFFFH. Furthermore, FFFFH is the setting value for simultaneous broadcast communication.

(c) Set the port numbers for the external device upon consulting a network administrator.

(d) The following table provides some precautions that should be observed when setting port numbers.
(□ in the diagram indicates a device and ○ indicates a port number.)

Status of connection establishment (○: Port (indicates port number))	Description of connection	Communication protocol	
		TCP	UDP
	When connecting to multiple external devices, set multiple port numbers for the local station.	○	○
	When connecting to multiple external devices, set single port number for the local station. (However, it is necessary to open a connection for each external device) This cannot be chosen when the local station is Unpassive.	○	×
	When connecting to multiple ports of an external device, set multiple port numbers for the Ethernet module.	○	○
	When connecting to multiple ports of an external device, set single port number for the Ethernet module. (However, it is necessary to open a connection for each external device port) This cannot be chosen when the local station is Unpassive.	○	×
	When connecting to the same ports of an external device, set multiple port number for the Ethernet module. (However, it is necessary to open a connection for each Ethernet module port)	○	○
	Setting multiple ports numbers for the same port of an external device and the Ethernet module is possible only when paring open is set.	○	○

Important

Parameters for connections that communicate by Passive open and UDP open must always be set from this screen when "Always wait for OPEN (communication possible at STOP time)" is selected in the Operational settings (see Section 4.7).

POINT

Set parameters according to the open method to be used for open connection.

Parameter \ Communication system / open system		TCP				UDP	
		Active		Passive			
		ARP function of external device		Un-passive	Full-passive	ARP function of external device	
		Yes	No			Yes	No
Communication address	Local station Port No.	○	○	○	○	○	○
	Destination IP address	○	○	×	○	○	○
	Destination Port No.	○	○	×	○	○	○
	Destination Ethernet address (＊2)	○ (＊1)	○	×	×	○ (＊1)	○

＊1 Use the default value ($FFFFFFFFFFFF_H$) or "0".

＊2 When using the "Open settings" of GX Developer, the default value is used.

When communicating with an external device without the ARP function, use the dedicated OPEN instruction and set the external device's Ethernet address in the control data.

5.6 Open Processing/Close Processing of the Connection

This section explains the open processing/close processing using sequence programs.

(1) Open processing

(a) The purpose of the open processing is to establish a connection with an external device in order to perform the following forms of data communication.
They can all be performed with an external device opened by the user.
- Communication using the MC protocol
- Sending/receiving using the fixed buffers (Procedure exists)
- Communication using the random access buffers

(b) When the following is set by the parameter settings of GX Developer, the open processing should be performed in a sequence program.

1) In the Operational settings (Section 4.7)
When "Do not wait for OPEN" is set in the "Initial timing setting".

2) In the open settings (Section 5.5)
When "Active" is set in the "OPEN system" setting.

(c) In order to perform the open processing, the initial processing must have been completed.

(d) A connection with an external device must be established (open processing) when communicating using either the MC protocol, fixed buffers, or random access buffers. (*1)
All the three types of data communication mentioned above can also by performed with an external device opened by the user.
*1 Since the Ethernet module recognizes the external device to communicate with by the IP address, the open processing is required for UDP communication.

(e) Up to a maximum of 16 connections can be opened to external devices. However, two buffers are required when communicating with the same external device using fixed buffer communication, so in this case the number of external devices that can be communicated with will be lower.

POINT
During communication using the MC protocol or random access buffers, if data communication still continues even after the PLC CPU in a station where an Ethernet module is installed has been placed in the STOP status, set "Always wait for OPEN (communication possible at STOP time)" under "Initial timing setting" (see 4.7, "Operation Settings").

(2) Close processing

(a) The purpose of the close processing is to disconnect (cancel) the connection with the external device that has been established by the open processing mentioned previously.

(b) The close processing is used when terminating a connection with an external device, changing an external device of a connection, changing communication conditions, etc.

(c) Perform the close processing for connections that have been established by the open processing using sequence programs.

(d) Determine the timing of close processing with the external device.

The examples in the following sections describe the procedures for establishing a connection from the Ethernet module to an external device and subsequently closing it again by open and close processing for connection number 1.

- TCP/IP Active open : See 5.6.1, "Active open processing/close processing".
- TCP/IP Passive open : See 5.6.2, "Passive open processing/close processing".
- UDP/IP Open : See 5.6.3, "UDP/IP open processing/close processing".

Important

Never execute the open/close processing to the same connection using the input/input signals and use together with dedicated OPEN/CLOSE instructions for OPEN/CLOSE processing. It may result in malfunctions.

POINT

Except when the close processing is requested, the open completion signal (address: applicable bit of 5000H) automatically turns off and the communication line is closed in the following cases:

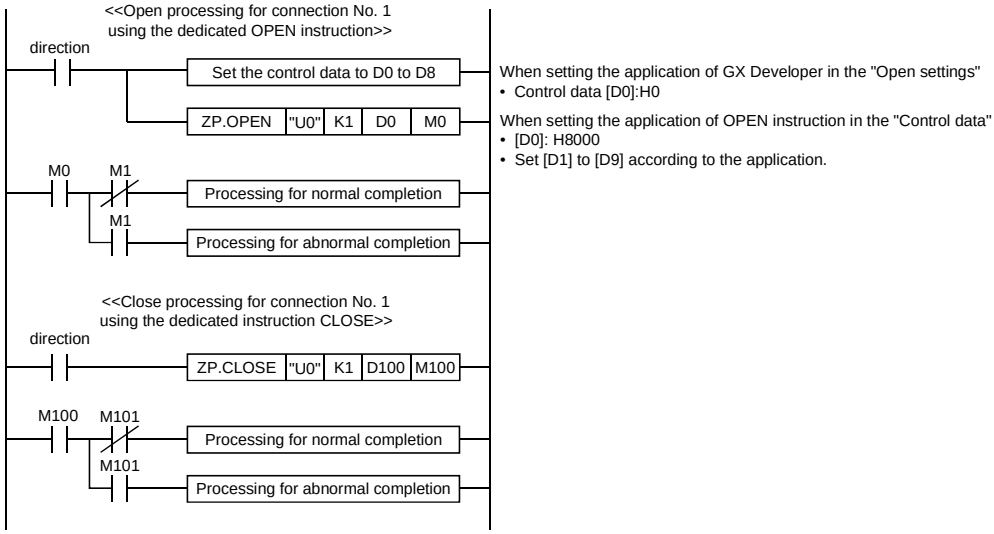
(1) When the existence confirmation function times out occur (see Section 5.5). (*1)

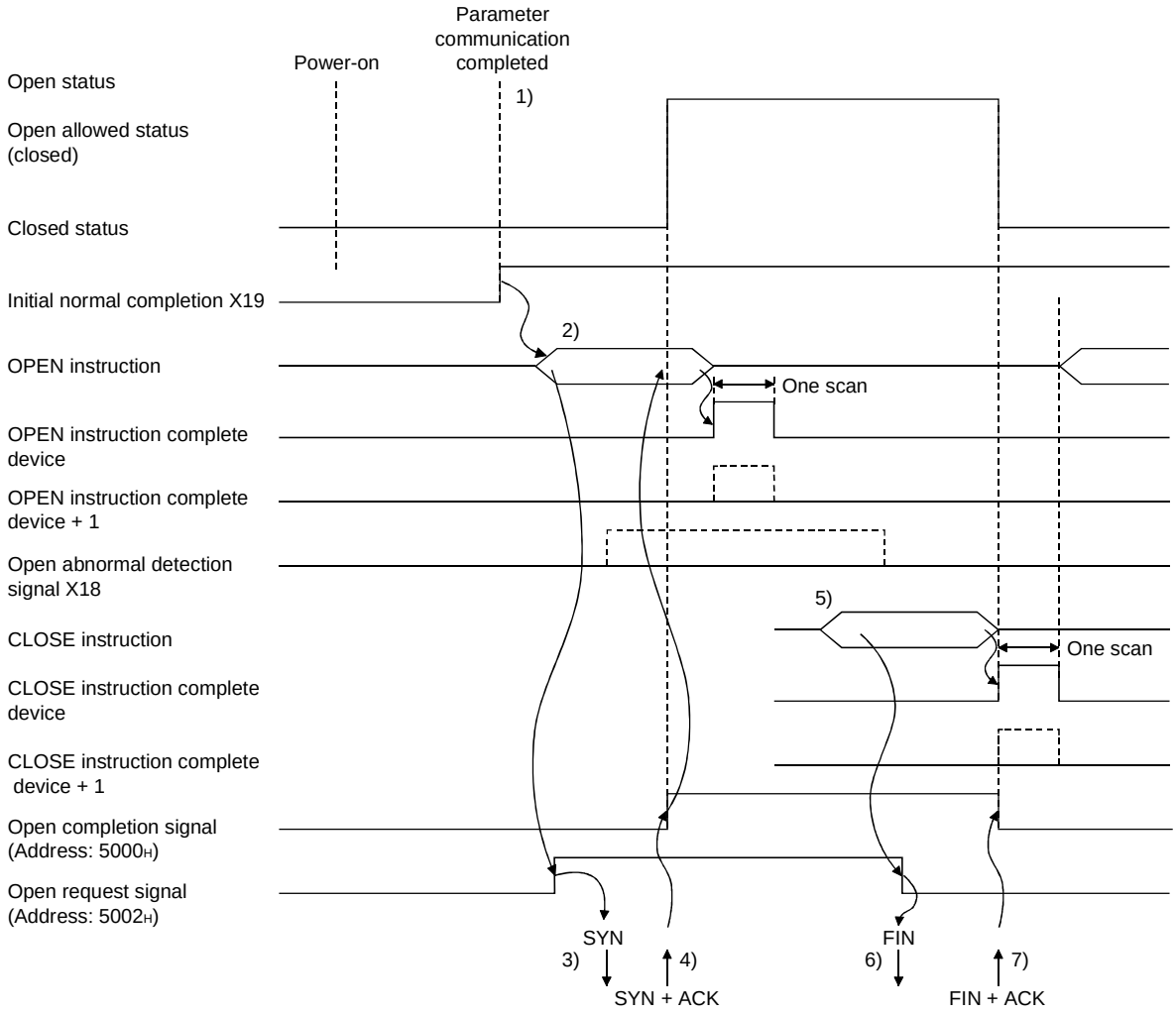
(2) When a close or the ABORT (RST) instruction is received from an external device.

(3) When the Active open request is received again from the external device in the open completion status of TCP.
Depending on its version, the Ethernet module operates as described below.
(a) Ethernet module whose first 5 digits of the serial No. are 05051 or later
After returning ACK to the external device, the Ethernet module closes the connection when the RST command is received from the external device.
(b) Ethernet module whose first 5 digits of the serial No. are 05049 or earlier
After sending the RST command, the Ethernet module closes the connection.
However, when receiving the Active open request again from the external device with a different IP address or port No., the Ethernet module only sends the RST command. (It does not close the connection.)

5.6.1 Active open processing/close processing

This section explains the procedure for opening and closing a connection with an external device from the Ethernet module.





5 - 45

1) After communicating the parameter settings, confirm the normal completion of the Ethernet module initial processing.
(Initial normal completion signal (X19): ON)

2) Start the open processing using the dedicated OPEN instruction.
(Open request signal (address: 5002_H … b0): ON)

3) The Ethernet module executes the open processing.
- Sends the open request (SYN).

4) When the open processing completes normally
- Open completion signal (address: 5000_H … b0) : ON
- OPEN instruction complete device : ON
- OPEN instruction complete device + 1 : OFF
- OPEN instruction complete status area (∗1) : 0000_H

Data communication is enabled.

When the open processing completes abnormally (∗2)
- Open completion signal : OFF
- OPEN instruction complete device : ON
- OPEN instruction complete device + 1 : ON
- The open error code is stored in the buffer memory. (∗3)
- OPEN instruction complete status area (∗1) : Value other than 0000_H
- Open abnormal detection signal (X18) : ON

5) Start the close processing using the dedicated CLOSE instruction.
(Open request signal: OFF)

6) The Ethernet module executes the close processing.
- Sends the close request (FIN).

7) When the close processing completes normally
- Open completion signal : OFF
- CLOSE instruction complete device : ON
- CLOSE instruction complete device + 1 : OFF
- CLOSE instruction complete status area (∗1) : 0000_H

When the close processing completes abnormally (∗4)
- Open completion signal : OFF
- CLOSE instruction complete device : ON
- CLOSE instruction complete device + 1 : ON
- CLOSE instruction complete status area (∗1) : Value other than 0000_H

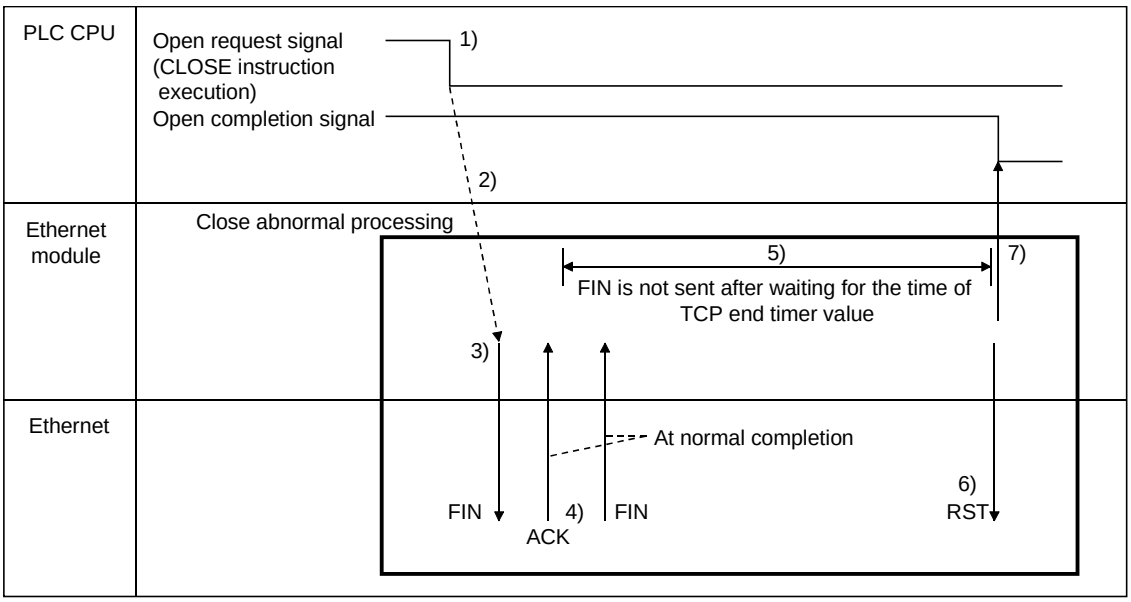
POINT
This example uses connection number 1 for explanation. Use the corresponding signals and bits for other connection numbers.

∗1 The end code at completion is stored in the dedicated instruction complete status area. For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

∗2 Processing when the open processing is abnormally completed (TCP)
When opening from the Ethernet module under a normal situation, if the Ethernet module sends a SYN, the external device returns an ACK and a SYN.
However, if the Ethernet module sends a SYN and then the external device returns a RST, the open abnormal completion (X18) is immediately turned on and open processing ends.

∗3 The open status and error code at abnormal end can be confirmed with the following buffer memory.
- Each connection open abnormal code area of the communication status storage area
  (Connection numbers 1 to 8; addresses: 78H to C7H)
  (Connection numbers 9 to 16; addresses: 5020H to 586FH)
- Error log area (addresses: E0H to 177H)
- Error codes stored in the open abnormal code area are cleared (n → 0) when the dedicated OPEN instruction is executed again.

∗4 Processing when the close processing is abnormally completed (TCP)
When closing normally from the Ethernet module, the Ethernet module sends a FIN request and the external device returns an ACK and a FIN.
However, if an ACK and a FIN are not returned because the external device is faulty, the Ethernet module forcefully disconnects the connection (send a RST message).



1) The open request signal is turned off by the dedicated CLOSE instruction.
2) The Ethernet module executes the close processing.
3) The Ethernet module sends a FIN request to the external device.
4) The external device sends back FIN and ACK messages in reply to the FIN request sent by the Ethernet module.
(When the reply is not returned, the Ethernet module sends the FIN request again.)
5) The Ethernet module waits for the external device to send an ACK and a FIN.
(The module waits for the amount of time set in the TCP end timer value. For details on how to set it, see Section 5.2, "Initial Settings".)
If the ACK and FIN messages are received at this point, it returns an ACK as in the normal processing.
6) If an ACK and a FIN are not received within the time designated in the TCP end timer value, an RST message is sent to the external device.
7) The Ethernet module determines that the close procedure is completed and turns off the open completion signal regardless of the status of the external device.

REMARK

(1) When the procedure above is performed, the Ethernet module determines that the closing of the external device is executed normally, thus the close processing result is not stored in the error log area.

(2) The procedure described above is a special function of the Ethernet module; it is not available for general TCP/IP protocols.

Program example

This example explains a program for open processing/close processing when Active open is selected in the Open system setting.

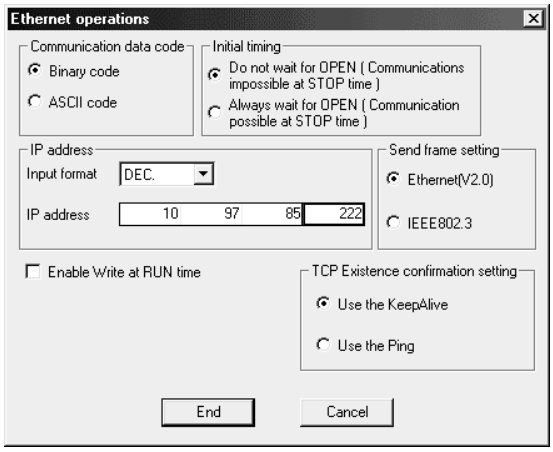
(1) Execution environment for the program example

(a) The Ethernet module is installed in the "0" slot of the basic base unit.

(b) The [Network Parameters Setting the number of MNET/10H Ethernet cards] settings are assumed to have been set using GX Developer as follows.

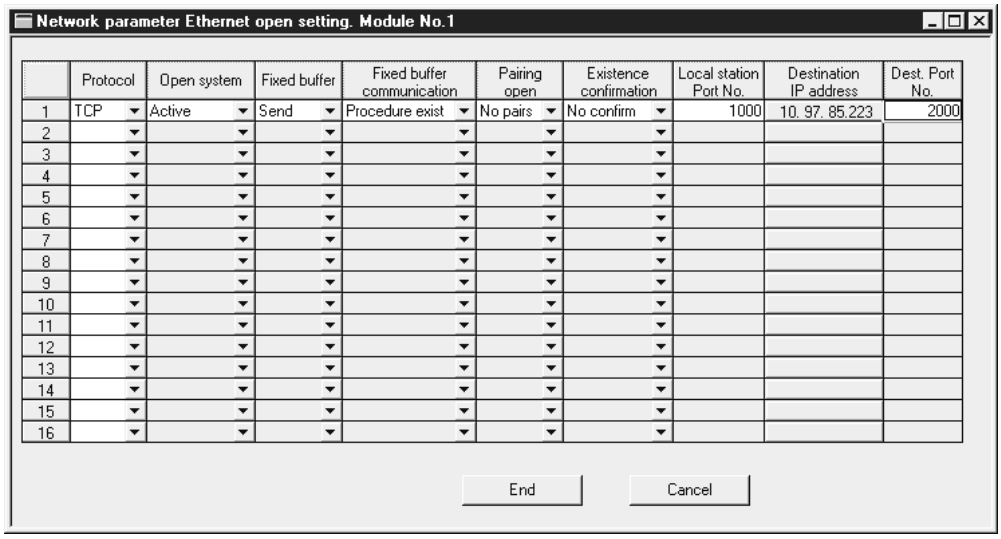
- Network type : Ethernet
- Starting I/O No. : 0000
- Network No. : 1
- Group No. : 1
- Station No. : 1

(c) The [Operational settings] are assumed to have been set using GX Developer as follows.



Local station IP address : 0A.61.55.DEH (10.97.85.222)

(d) The [Open settings] parameters are assumed to have been performed using GX Developer as follows.



Local station Port No. : 1000H
Destination IP address : 0A.61.55.DFH (10.97.85.223)
Destination Port No. : 2000H

(e) The following contact signals are used in the program.

Connection No. 1 open completion signal	: M0
Connection No. 1 open request signal	: M20
Connection No. 1 OPEN instruction control data	: Stored in D100 to D109
Connection No. 1 CLOSE instruction control data	: Stored in D200 and D201

(f) The area enclosed with ⋮ in the program example should be used when the [Open settings] Ethernet module parameters are not set for GX Developer.
This part of the program is not required when the [Open settings] parameters are used for GX Developer.

(g) For details on the dedicated OPEN instruction, see Chapter 10, "Dedicated Instructions".

(2) Outline of the program example

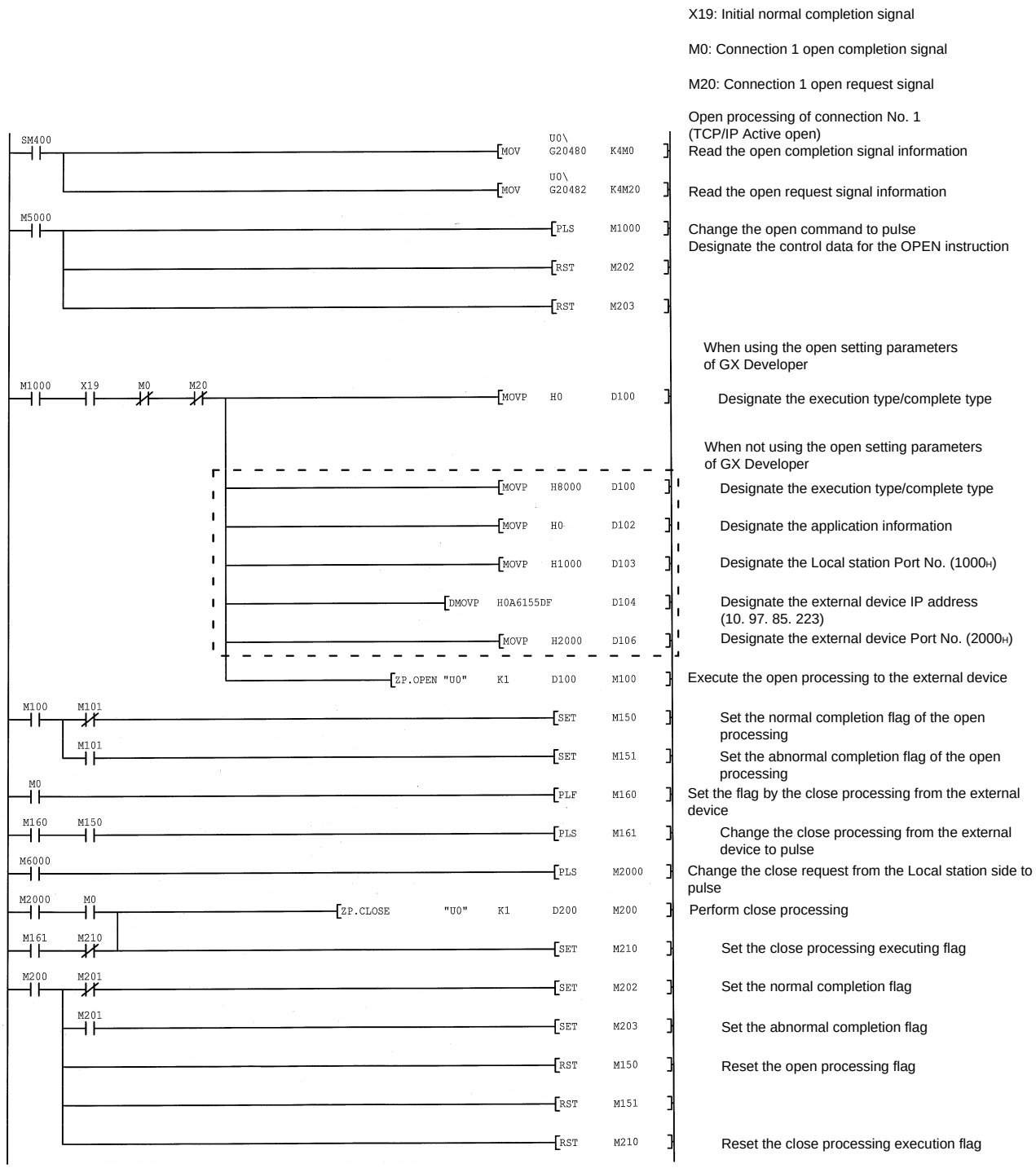
(a) After each parameter is set from GX Developer and write to PLC CPU, restart the PLC CPU and confirm the completion of the initial processing.

(b) The Ethernet module performs the open processing for connection No. 1 to the external device set in the [Open settings] or control data.

(c) The close processing for connection No. 1 is performed according to the close instruction to the Ethernet module or the close request from the external device.

REMARK

The "U0\G20480" and "U0\G20482" codes shown in the program designate the following areas in the buffer memory.
U0\G20480: Open completion signal storage area (address: 5000_H (20480))
U0\G20482: Open request signal storage area (address: 5002_H (20482))


X19: Initial normal completion signal
M0: Connection 1 open completion signal
M20: Connection 1 open request signal
Open processing of connection No. 1 (TCP/IP Active open)
SM400
MOV U0\G20480 K4M0
Read the open completion signal information
MOV U0\G20482 K4M20
Read the open request signal information
M5000
PLS M1000
Change the open command to pulse
RST M202
Designate the control data for the OPEN instruction
RST M203
When using the open setting parameters of GX Developer
M1000 X19 M0 M20
MOVP H0 D100
Designate the execution type/complete type
When not using the open setting parameters of GX Developer
MOVP H8000 D100
Designate the execution type/complete type
MOVP H0 D102
Designate the application information
MOVP H1000 D103
Designate the Local station Port No. (1000H)
DMOVP H0A6155DF D104
Designate the external device IP address (10. 97. 85. 223)
MOVP H2000 D106
Designate the external device Port No. (2000H)
ZP.OPEN "U0" K1 D100 M100
Execute the open processing to the external device
M100 M101
SET M150
Set the normal completion flag of the open processing
M101
SET M151
Set the abnormal completion flag of the open processing
M0
PLF M160
Set the flag by the close processing from the external device
M160 M150
PLS M161
Change the close processing from the external device to pulse
M6000
PLS M2000
Change the close request from the Local station side to pulse
M2000 M0
ZP.CLOSE "U0" K1 D200 M200
Perform close processing
M161 M210
SET M210
Set the close processing executing flag
M200 M201
SET M202
Set the normal completion flag
M201
SET M203
Set the abnormal completion flag
RST M150
Reset the open processing flag
RST M151
RST M210
Reset the close processing execution flag

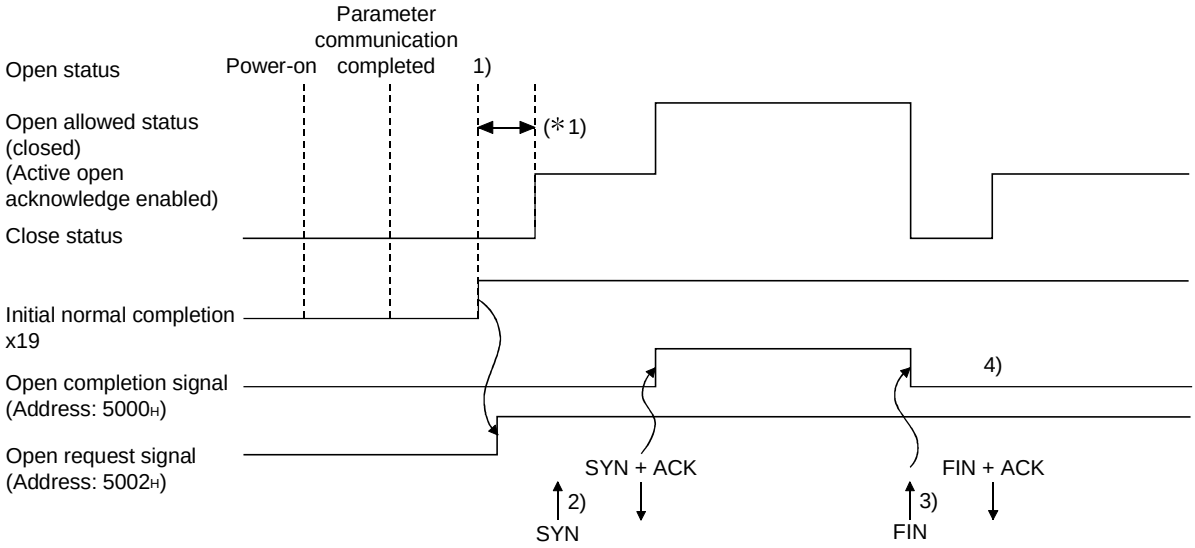
5 - 51

5.6.2 Passive open processing/close processing

This section explains the procedure for opening and closing a connection with the Ethernet module from an external device.
The operations of Passive open processing/close processing differ depending on whether "Always wait for OPEN" or "Do not wait for OPEN" is selected, as shown in this section.

(1) When "Always wait for OPEN" is selected in the operational setting

When [Operational setting] - [Initial timing setting] - [Always wait for OPEN (Communication possible at STOP time)] is selected from GX Developer, the open processing/close processing is performed as explained below.
In this case, sequence programs for open processing and close processing are not required because the Ethernet module keeps the connection in the always wait for the OPEN status according to the [Open settings] parameter setting.
For detail on the [Open settings] parameter, see Section 5.5, "Open Settings".



1) After the parameters are sent, the normal completion of the Ethernet module initial processing is confirmed (Initial normal completion signal (X19): ON)
   After the initial processing is normally completed, the connection is placed in the open allowed status and Ethernet module waits for an open request from the external device.
2) The Ethernet module starts the open processing upon receiving the open request (SYN) from the external device.
   When the open processing is normally completed, the open completion signal (address: 5000H… b0) turns on and data communication is enabled.
3) The Ethernet module starts the close processing upon receiving the close request (FIN) from the external device.
   When the close processing is completed, the open completion signal turns off and data communication is disabled.
4) After the Ethernet module's internal processing is completed, the connection returns to the open acknowledge enabled status.

*1 An open request (SYN) received after the normal completion of an initial processing and before the Ethernet module is placed in the open acknowledge enabled status generates an error, and the Ethernet module sends a connection forced close (RST).

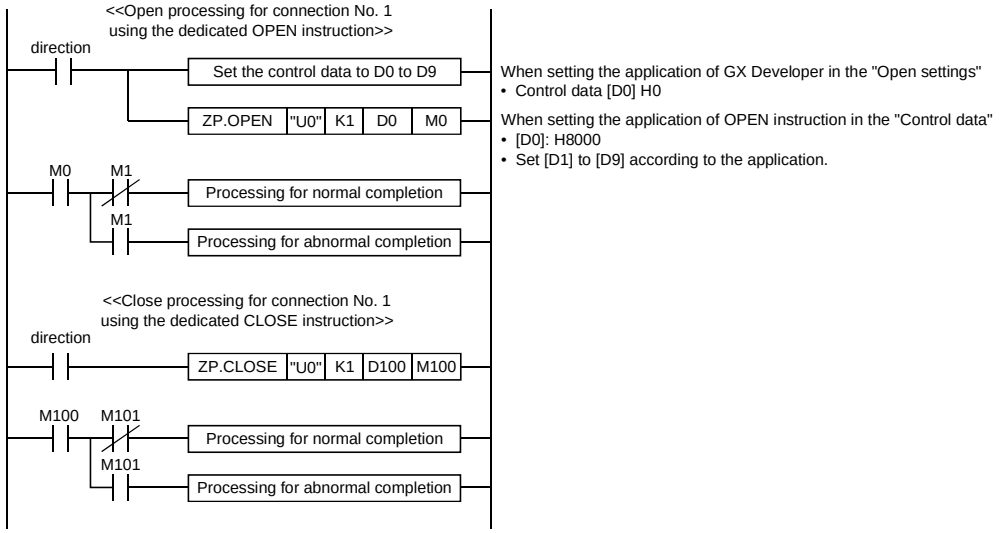
REMARK

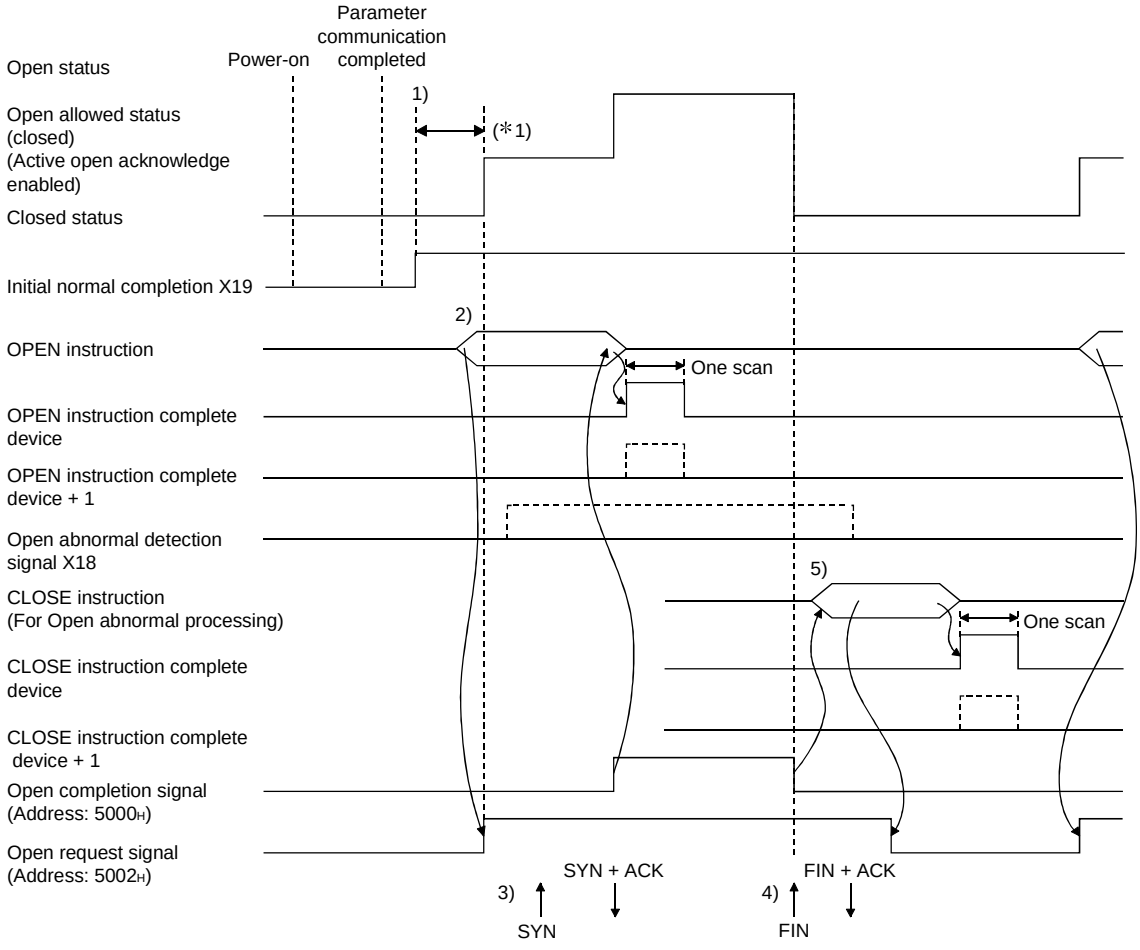
For Passive open connections for which [Always wait for OPEN (Communication possible at STOP time)] is selected in [Operational settings], the connection open/close processing of Ethernet module side is performed according to the open/close request from the external device.
When the close processing is performed from the Ethernet module side using the dedicated CLOSE instruction, the applicable connection will not return to the open acknowledge enabled status after the close processing. (It requires the same open processing as for Passive open connection for which "Do not wait for OPEN (Communications impossible at STOP time)" is selected).

(2) When "Do not wait for OPEN" is selected in the operational setting

When [Operational setting] - [Initial timing setting] – [Do not wait for OPEN (communications impossible at STOP time)] is selected from GX Developer, the open processing/close processing is performed as explained below.

Since open processing/close processing is executed by a sequence program, an external device can be changed while the connection is established.





1) After communicating the parameter settings, confirm the normal completion of the Ethernet module initial processing.
(Initial normal completion signal (X19): ON)

2) Start the open processing using the dedicated OPEN instruction.
(Open request signal (address: 5002H … b0): ON)

3) The Ethernet module starts the open processing upon receiving the open request (SYN) from the external device.

At normal completion
- Open completion signal (address: 5000H … b0) : ON
- OPEN instruction complete device : ON
- OPEN instruction complete device + 1 : OFF
- OPEN instruction complete status area (*2) : 0000H

Data communication is enabled.

At abnormal completion
- Open completion signal : OFF
- OPEN instruction complete device : ON
- OPEN instruction complete device + 1 : ON
- The open error code is stored in the buffer memory
- OPEN instruction complete status area (*2) : Value other than 0000H
- Open abnormal detection signal (X18) : ON

4) The Ethernet module starts the close processing upon receiving the close request (FIN) from the external device.
When the close processing is completed, the open completion signal turns off and data communication is disabled.

5) Start the close processing using the dedicated CLOSE instruction.
(Open request signal: OFF)

At normal completion of internal processing
- Open completion signal : OFF
- CLOSE instruction complete device : ON
- CLOSE instruction complete device + 1 : OFF
- CLOSE instruction complete status area (*2) : 0000H

At abnormal completion of internal processing
- Open completion signal : OFF
- CLOSE instruction complete device : ON
- CLOSE instruction complete device + 1 : ON
- CLOSE instruction complete status area (*2) : Value other than 0000H

POINT

This example uses connection number 1 for explanation. Use the corresponding signals and bits for other connection numbers.

*1 An open request (SYN) received after the normal completion of an initial processing and before the Ethernet module is placed in the open acknowledge enabled status generates an error, and the Ethernet module sends a connection forced close (RST).
*2 The end code at completion is stored in the dedicated instruction complete status area. For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

REMARK

(1) If the settings of the connection need modifying, the modifications should be done before executing the dedicated OPEN instruction.

(2) Once open processing is executed, an open request cannot be canceled before the open processing completes.
Perform close processing (CLOSE instruction) after the open processing has completed.

Program example

This example explains a program for open processing/close processing when Unpassive open is selected in the open system setting.

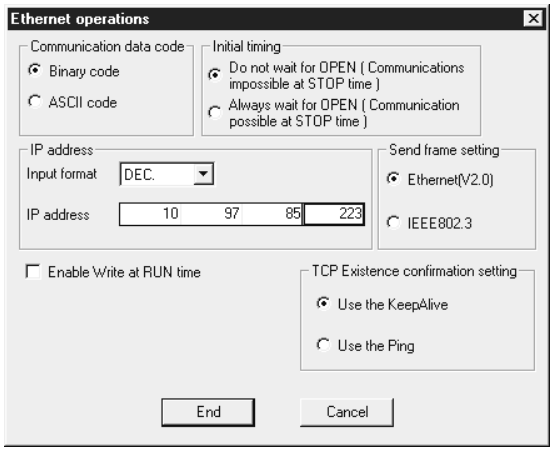
(1) Execution environment for the program example

(a) The Ethernet module is installed in the "0" slot of the basic base unit.

(b) The [Network Parameters Setting the number of MNET/10H Ethernet cards] settings are assumed to have been set using GX Developer as follows.

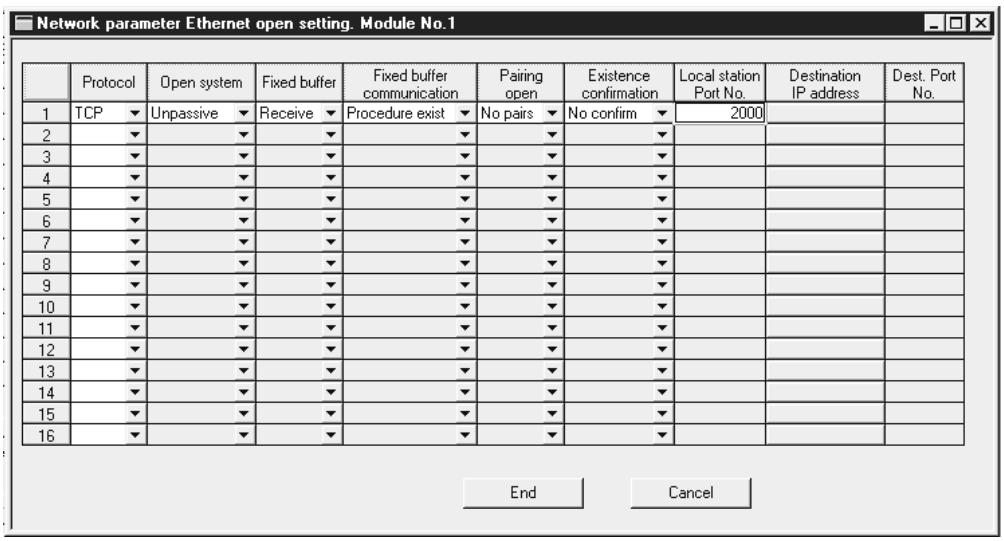
- Network type : Ethernet
- Starting I/O No. : 0000
- Network No. : 1
- Group No. : 1
- Station No. : 2

(c) The [Operational settings] parameters are assumed to have been set using GX Developer as follows.



Local station IP address : 0A.61.55.DF$_H$ (10.97.85.223)

(d) The [Open settings] are assumed to have been set using GX Developer as follows.



Local station Port No. : 2000$_H$

(e) The following contact signals are used in the program.

Connection No. 1 open completion signal	: M0
Connection No. 1 open request signal	: M20
Connection No. 1 OPEN instruction control data	: Stored in D100 to D109
Connection No. 1 CLOSE instruction control data	: Stored in D200 and D201

(f) The area enclosed with ⋮ in the program example should be used when the [Open settings] Ethernet module parameter are not set for GX Developer.
This part of the program is not required when the [Open settings] parameters are used for GX Developer.

(g) For details on the dedicated OPEN instruction, see Chapter 10, "Dedicated Instructions".

(2) Outline of the program example

(a) After each parameter is set from GX Developer and write to PLC CPU, restart the PLC CPU and confirm the completion of the initial processing.

(b) The open processing for connection No. 1 of the Ethernet module is performed.
After the completion of the open processing, connection No. 1 waits for the open request from the external device.

(c) The close processing for connection No. 1 is performed according to the close instruction to the Ethernet module or the close request from the external device.

REMARK

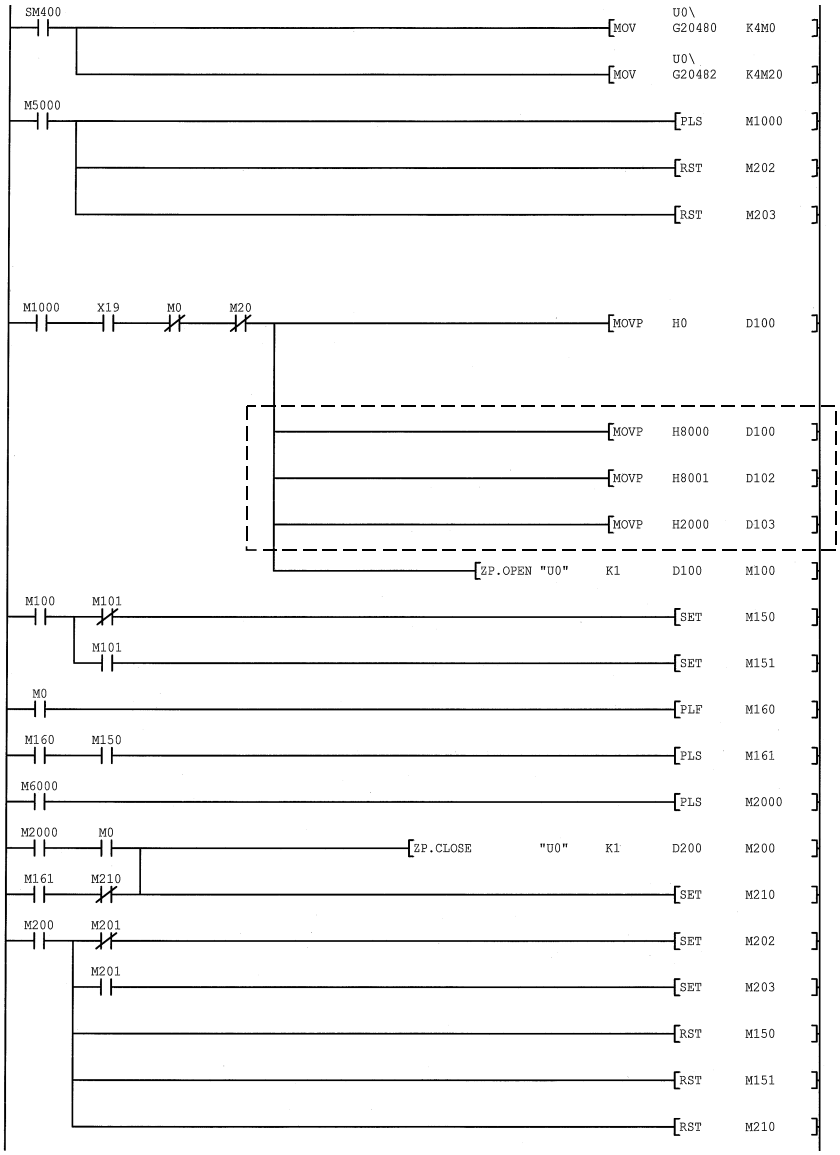
The "U0\G20480" and "U0\G20482" codes shown in the program designate the following areas in the buffer memory.
U0\G20480: Open completion signal storage area (address: 5000_H (20480))
U0\G20482: Open request signal storage area (address: 5002_H (20482))

X19: Initial normal completion signal

M0: Connection 1 open completion signal

M20: Connection 1 open request signal

Open processing of connection No. 1
(TCP/IP Unpassive open)



Read the open completion signal information

Read the open request signal information

Change the open command to pulse

Designate the control data for the OPEN instruction

When using the open setting parameters of GX Developer

Designate the execution type/complete type

When not using the open setting parameters of GX Developer

Designate the execution type/complete type

Designate the information of the application

Designate the Local station Port No. (2000_H)

Execute the open processing to the external device

Set the normal completion flag of the open processing

Set the abnormal completion flag of the open processing

Set the flag when the close processing is performed from the external device

Change the close processing from the external device to pulse

Change the close request from the Local station side to pulse

Perform close processing

Set the close processing executing flag

Set the normal completion flag

Set the abnormal completion flag

Reset the open processing flag

Reset the close processing execution flag

5 - 59

5.6.3 UDP/IP open processing/close processing

The following explains the UDP/IP open processing.

The operations of the open processing/close processing in UDP/IP differ depending on whether "Always wait for OPEN" or "Do not wait for OPEN" is selected in the settings using GX Developer, [Operational settings] – [Initial settings], as shown below.

(1) When "Always wait for OPEN" is selected in the operational settings (Communications possible at STOP time)

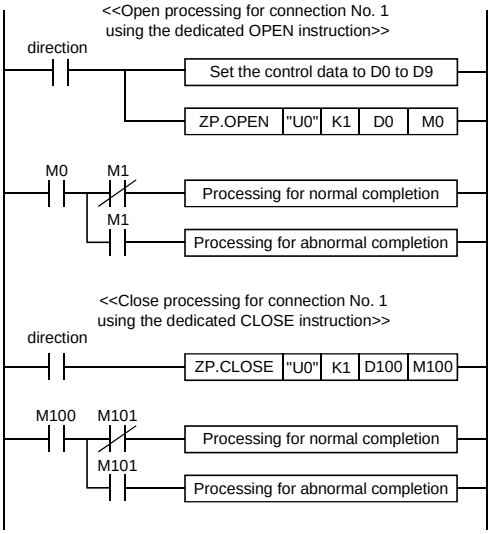
According to the [Open settings] with GX Developer, a connection for which the UDP/IP communication is selected is established automatically after the Ethernet module installed station has been restarted, and the data transmission/reception is enabled.
Sequence programs for open processing and close processing are not required.
For details on the [Open settings] parameter, see Section 5.5, "Open Settings".

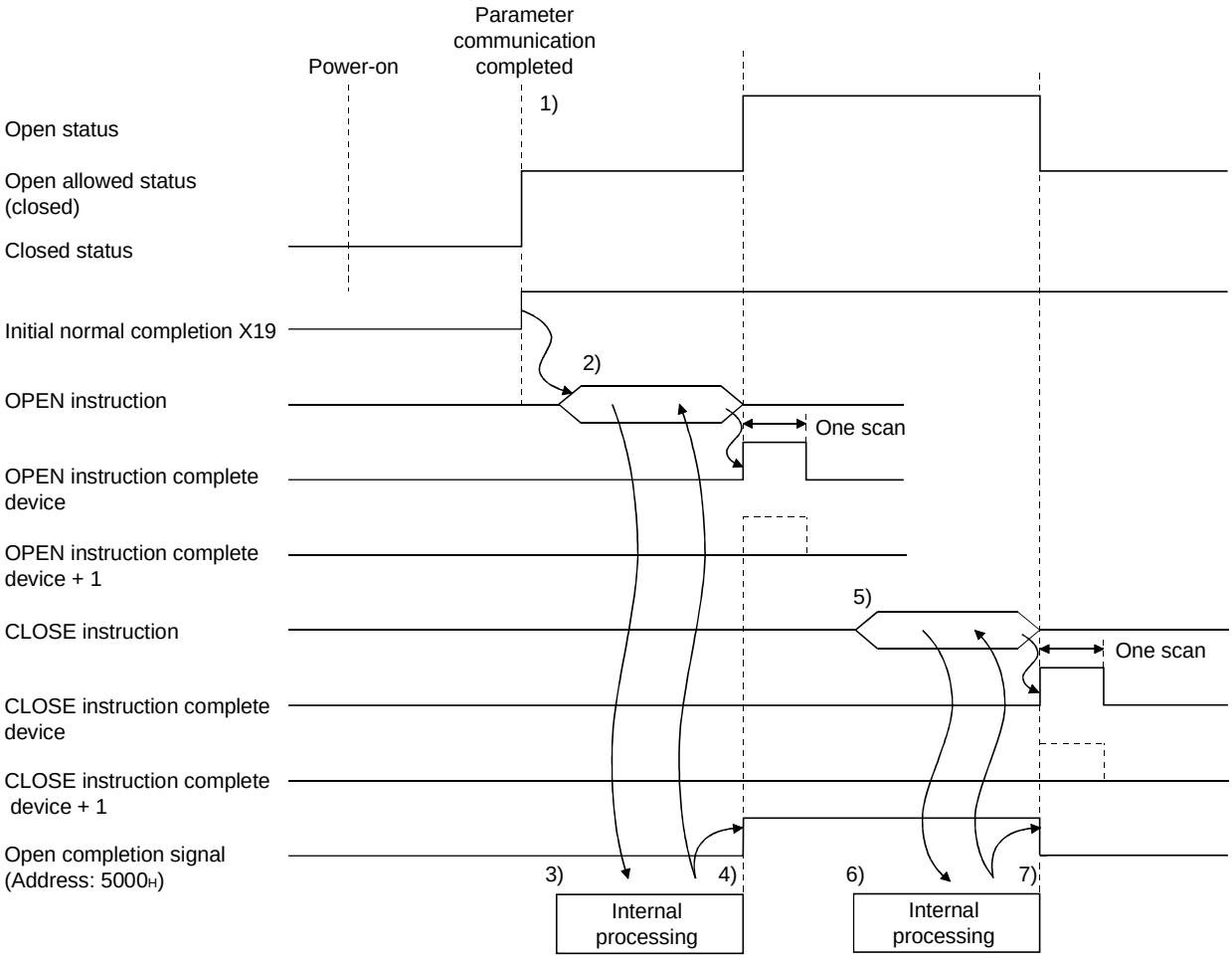
REMARK

Even when "Always wait for OPEN (Communication possible at STOP time)" is selected at "Operational settings", if there is a dedicated OPEN instruction from the Ethernet mudule, open processing using the CLOSE instruction and close processing, there is a need to perform all open processing and close processing after the relevant connection using the sequence program.
If "Do not wait for OPEN (Communication impossible at STOP time) is selected, there is a need to perform open processing and close processing in the same manner as a connection.

(2) When "Do not wait for OPEN" is selected in the operational settings (Communications impossible at STOP time)

In this case, the open processing and close processing in the sequence program shown in the next page are required. Data transmission and reception are enabled after the open processing is normally completed.
Perform the open processing and close processing using the applicable dedicated instructions.
For more details, see Chapter 10, "Dedicated Instructions".


<<Open processing for connection No. 1
using the dedicated OPEN instruction>>
direction
Set the control data to D0 to D9
ZP.OPEN
"U0"
K1
D0
M0
When setting the application of GX Developer in the "Open settings"
• Control data [D0]: H0
When setting the application of OPEN instruction in the "Control data"
• [D0]: H8000
• Set [D1] to [D9] according to the application.
M0
M1
Processing for normal completion
M1
Processing for abnormal completion
<<Close processing for connection No. 1
using the dedicated CLOSE instruction>>
direction
ZP.CLOSE
"U0"
K1
D100
M100
M100
M101
Processing for normal completion
M101
Processing for abnormal completion


Power-on
Parameter communication completed
1)
Open status
Open allowed status (closed)
Closed status
Initial normal completion X19
2)
OPEN instruction
One scan
OPEN instruction complete device
OPEN instruction complete device + 1
5)
CLOSE instruction
One scan
CLOSE instruction complete device
CLOSE instruction complete device + 1
Open completion signal (Address: 5000H)
3)
4)
6)
7)
Internal processing
Internal processing

5 - 61

1) After communicating the parameter settings, confirm the normal completion of the Ethernet module initial processing.
(Initial normal completion signal (X19): ON)

2) Start the open processing using the dedicated OPEN instruction.
(Open request signal (address: 5002_H … b0): ON)

3) The Ethernet module executes the open processing. (Internal processing only)

4) When the open processing completes normally
- Open completion signal (address: 5000_H … b0) : ON
- OPEN instruction complete device : ON
- OPEN instruction complete device + 1 : OFF
- OPEN instruction complete status area (*1) : 0000_H

Data communication is enabled.

When the open processing completes abnormally
- Open completion signal : OFF
- OPEN instruction complete device : ON
- OPEN instruction complete device + 1 : ON
- The open error code is stored in the buffer memory
- OPEN instruction complete status area (*1) : Value other than 0000_H
- Open abnormal detection signal (X18) : ON

5) Start the close processing using the dedicated CLOSE instruction.
(Open request signal: OFF)

6) The Ethernet module executes the close processing. (Internal processing only)

7) When the close processing completes normally
- Open completion signal : OFF
- CLOSE instruction complete device : ON
- CLOSE instruction complete device + 1 : OFF
- CLOSE instruction complete status area (*1) : 0000_H

When the close processing completes abnormally
- Open completion signal : OFF
- CLOSE instruction complete device : ON
- CLOSE instruction complete device + 1 : ON
- CLOSE instruction complete status area (*1) : Value other than 0000_H

*1 The end code at completion is stored in the dedicated instruction complete status area. For details on the dedicated instructions, see Chapter 10, “Dedicated Instructions“.

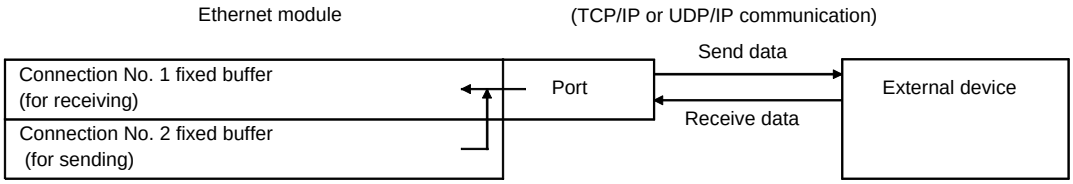
5.7 Pairing Open

The following explains communication using the paring open method via the Ethernet module.

5.7.1 Pairing open

Pairing open is an opening method for establishing a connection in which the local station and the external device use a single port for each, by pairing the receiving and sending connections through fixed buffer communication (both the procedure exists and no procedure are allowed) of the Ethernet module.
By designating the paring open method, data communication can be performed with two connections by performing open processing for only one port.
Communications using the MC protocol and the random access buffers can also be performed using these paring-opened connections.
The procedure for performing the open/close processing for pairing open is explained below.

[Example]



POINT

(1) When setting the pairing open method, the fixed buffer of the applicable connection number (for receiving only) and the fixed buffer of the immediately succeeding connection number (for sending only) are paired.
For the applicable connection (for receiving only), designate in ranges of connection No. 1 to 7 and 9 to 15.

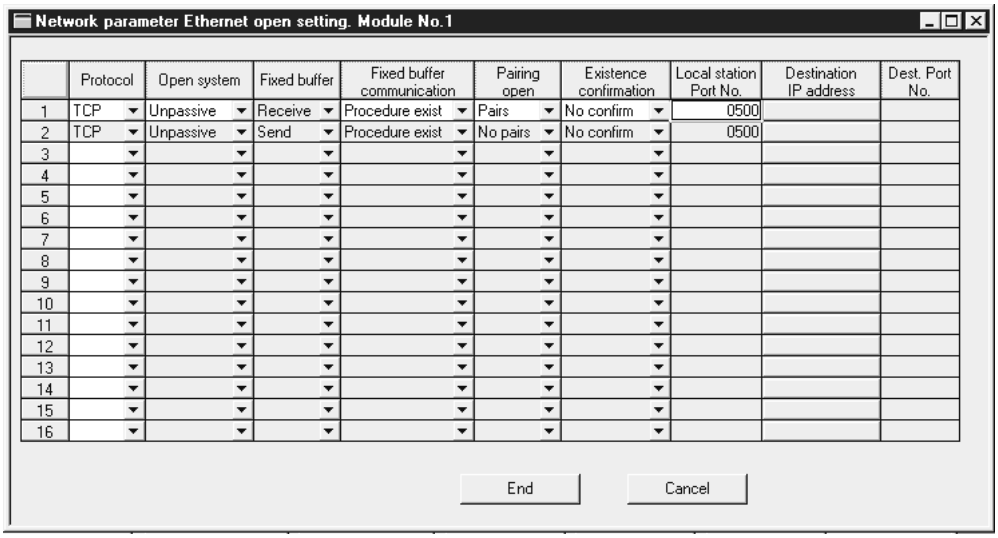
(2) The range of external devices that can be communicated by the pairing open method are limited to devices in the Ethernet to which the Ethernet module is connected and devices connected with the router relay function (see Section 5.3, "Router Relay Parameter").

(3) By the open/close processing of the applicable connection (for receiving only) for which the pairing open method has been set, the open/close processing of the next connection (for sending only) will automatically be performed.

5.7.2 Example of pairing open settings from GX Developer

This section explains the settings using GX Developer in order to communicate in the pairing open method by giving an example.
The screen below shows an example of the settings under the following conditions.

- Connection No. 1 and 2 are used.
- The Ethernet module port number is 0500H.
- The Unpassive open system is used.



(1) Protocol

Both "TCP/IP" and "UDP/IP" are allowed.

(2) Open system

All the open systems - "Active", "Unpassive", and "Fullpassive" - can be set.

(3) Fixed buffer

In the open pairing, the applicable connection No. and the subsequent connection No. are paired. Set the applicable connection No. to "Receive" and the next connection number to "Send".

(4) Fixed buffer communication

Both "Procedure exist" and "No procedure" can be selected.

(5) Pairing open

Set the receiving connection to "Pairs" and the sending connection to "No pairs".

(6) Existence confirmation

If existence confirmation is going to be executed, set the receiving connection to "Confirm" and the sending connection to "No confirm".
If existence confirmation is not going to be executed, select "No confirm" for both.

(7) Local station Port No.

Set this for the receiving connection for receiving only. (Setting is not required for the sending connection.)
Set the port number upon consulting a network administrator.

(8) Destination IP address

(a) If the Unpassive system is chosen
Setting is not required.

(b) If either Active or Fullpassive system is chosen
Setting is required.
Set for the receiving connection only upon consulting a network administrator.

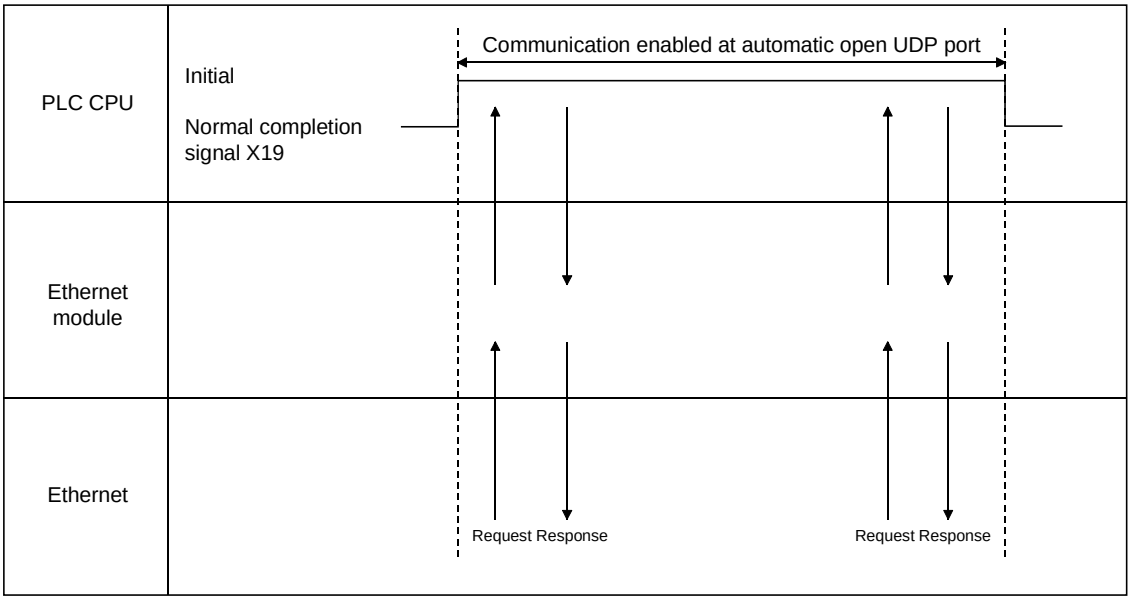
(9) Destination Port No.

(a) If the Unpassive system is chosen
Setting is not required.

(b) If either Active or Fullpassive system is chosen
Setting is required.
Set for the receiving connection only upon consulting a network administrator.

5.8 Automatic Open UDP Port

Normally, when communicating with an external device, it is necessary to establish a communication line with the external device when starting the data communication and to disconnect the communication line with the external device again when terminating the data communication following the arrangement with the external device.
The automatic open UDP port automatically opens/closes under the following conditions.
The port is placed in the communication enabled status after the initial processing is completed and communication can be performed without programming and regardless of the open status of connection No. 1 to 16.



(1) Open and close timing of automatic open UDP port

(a) Open timing
The port opens automatically according to the parameters registered by the user after the Ethernet module completes the initial processing, and the communication line is connected.

(b) Close timing
It automatically closes again when the station in which the Ethernet module is installed is reset/powered off.

(2) Functions that accommodate data communication using the automatic open UDP port

(a) Communication from an external device
1) Reading/writing data from/to the PLC CPU (the port number is designated by the user; default: 1388H (5000)) when communicating using the MC protocol (QnA compatible 3E frame command).
2) Regardless of the communication data code setting for the GX Developer (See Section 4.7). Communicate with the binary code.

(b) Communication from the station in which the Ethernet module is installed
In this case, it is possible to communicate using data link instructions (using port numbers of the operating system of the Ethernet module).

POINT

(1) The Ethernet module enables communication through the automatic open UDP port numbers after the initial processing is normally completed, and waits for communication requests to the local station Ethernet module. (Automatic open)

(2) It acknowledges and processes requests from anywhere as long as they are requests addressed to the Ethernet module itself.

(3) When a communication request is acknowledged, the applicable port number is occupied until the processing is completed. Even if the next communication request is acknowledged during this time, the communication processing has to wait.

(4) The automatic open UDP port can be used in communication between Ethernet modules using the MELSECNET/H, MELSECNET/10 relay communication function.

(5) Re-initial processing is required to change the port number of the automatic open UDP port. (Refer to Section 5.2.3 for re-initial processing.)

(3) Accessible range

The accessible range differs depending on the data communication function.

Data communication function	Accessible range	Reference section
Communication using the MC protocol (QnA comparatible 3E frame)	• The QCPU of the Ethernet module installed station • PLC CPU in the network system containing the Ethernet module installed station	—
Data link instruction	• PLC CPU in the Ethernet to which the Ethernet module is connected • PLC CPU in the Ethernet connected via a router • PLC CPU in the network system containing the Ethernet module installed station	Chapter 4 of Application

(4) Maximum size of data per communication

The maximum size of data per communication differs depending on the data communication function.

Data communication function	Maximum size of data per communication	Reference section
Communication using the MC protocol (QnA comparatible 3E frame)	The number of data that can be designated with the commands for the QnA compatible 3E frame	Chapter 6
Data link instruction	The number of data that can be designated with the commands for data link	Chapter 4 of Application

5.9 Corresponding with the QCPU Remote Password Function

The remote password function is one of the QCPU functions for the purpose of preventing improper access to the QCPU from users at the remote location.
The remote password function can be used by setting a remote password in the QCPU.
This section explains Ethernet module data communication in relation to the QCPU remote password function.

The remote password function is a function that has been added to the QCPU as a means of preventing improper access (such as destroying a program or data) from an external device. However, this function cannot completely prevent improper access.
The user should incorporate his/her own safeguards when it is need to keep the security of the PLC system against improper access from an external device.
The company cannot assume any responsibility for any problems that may arise from system troubles caused by improper access.

* Examples of measures to prevent improper access
  - Establish a firewall.
  - Set up a personal computer as a relay station and control the relay of sending/receiving data using an application program.
  - Set up an access controllable external device as a relay station.
    Please consult with a network connection vendor or equipment sales vendor regarding access controllable external devices.

5.9.1 Data communication when a remote password is set

This section explains the use and setting of the QCPU remote password function and data communication between the external device and the QCPU when a remote password has been set.

(1) Overview of the remote password function

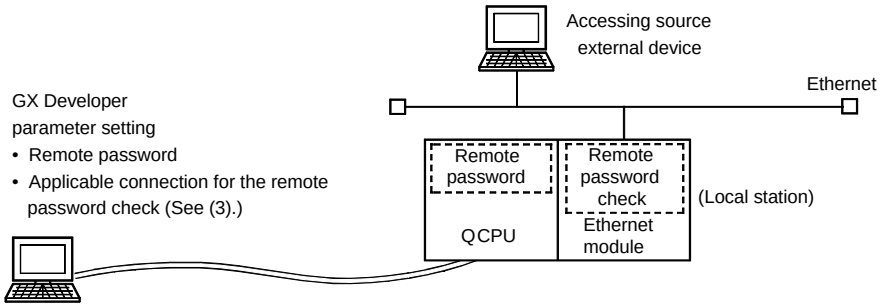
(a) Remote password function

1) The remote password function enables/disables access from the external device to the QCPU via the following modules:
   - Ethernet module
   - Q series C24

2) When the remote password function is enabled for the Ethernet module, the remote password check is performed for access from the external device to the following connections:
   Users can select desired connections to which the remote password check is to be applied.
   - User connection No.1 to 16
   - Auto open UDP port
   - FTP transmission port (TCP/IP)
   - GX Developer transmission port (TCP/IP)
   - GX Developer transmission port (UDP/IP)
   - HTTP port

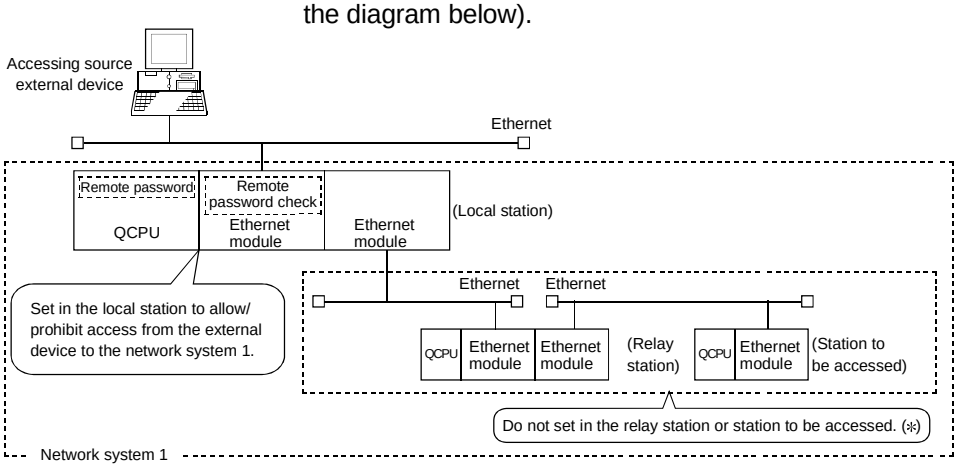
(b) Station that the remote password and remote password check are set

1) In the case of single network system



2) In the case of multiple network system

Set in the QCPU station that serves as the entrance to the PLC system as viewed from the external device (the local station QCPU in the diagram below).



* Setting in a station other than the QCPU station that serves as the entrance to the PLC system (relay station and station to be accessed in the above diagram), access to other stations beyond the set station is prohibited. (See (3).)

5 - 69

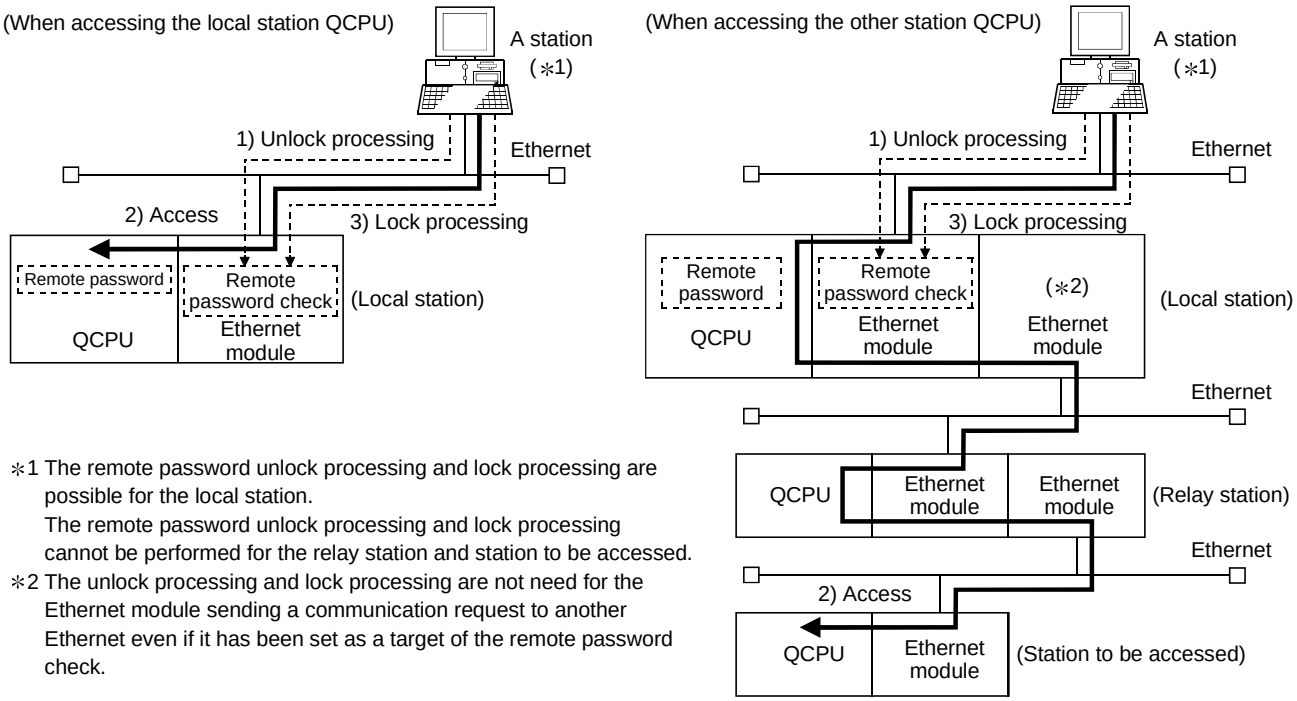
(2) Processing for allowing/prohibiting access to the PLC from the external device

1) Access allow processing (unlock processing)
- To access the specified QCPU, the external device performs the remote password unlock processing for the Ethernet module (*) of the directly connected station (local station).
- If the unlock processing has not been performed, access to the specified station is prohibited by the remote password check performed by the Ethernet module (*) that has received a communication request. (See (3).)
- All data receiving before the unlock processing will be processed as an error.

* The Ethernet module of the QCPU station for which the remote password is set.

2) Access processing
- Normal completion of the remote password unlock processing allows to access the specified station.
- Perform any access.

3) Access prohibit processing (lock processing)
- The remote password lock processing is performed from the external device to disable any further access when access to the specified station has been completed.



*1 The remote password unlock processing and lock processing are possible for the local station.
The remote password unlock processing and lock processing cannot be performed for the relay station and station to be accessed.

*2 The unlock processing and lock processing are not need for the Ethernet module sending a communication request to another Ethernet even if it has been set as a target of the remote password check.

POINT

(1) The remote password unlock processing and lock processing are performed for the only Ethernet module in the local station directly connected to the external device.
The remote password unlock processing and lock processing cannot be performed for an Ethernet module in any other station (relay station and station to be accessed).

(2) The remote password unlock processing and lock processing are performed from the external device using dedicated instructions for MC protocol communication. (They are performed with dedicated FTP commands when using the file transfer (FTP server) function. The remote password is input with the dialog box when using the Web function or GX Developer.)

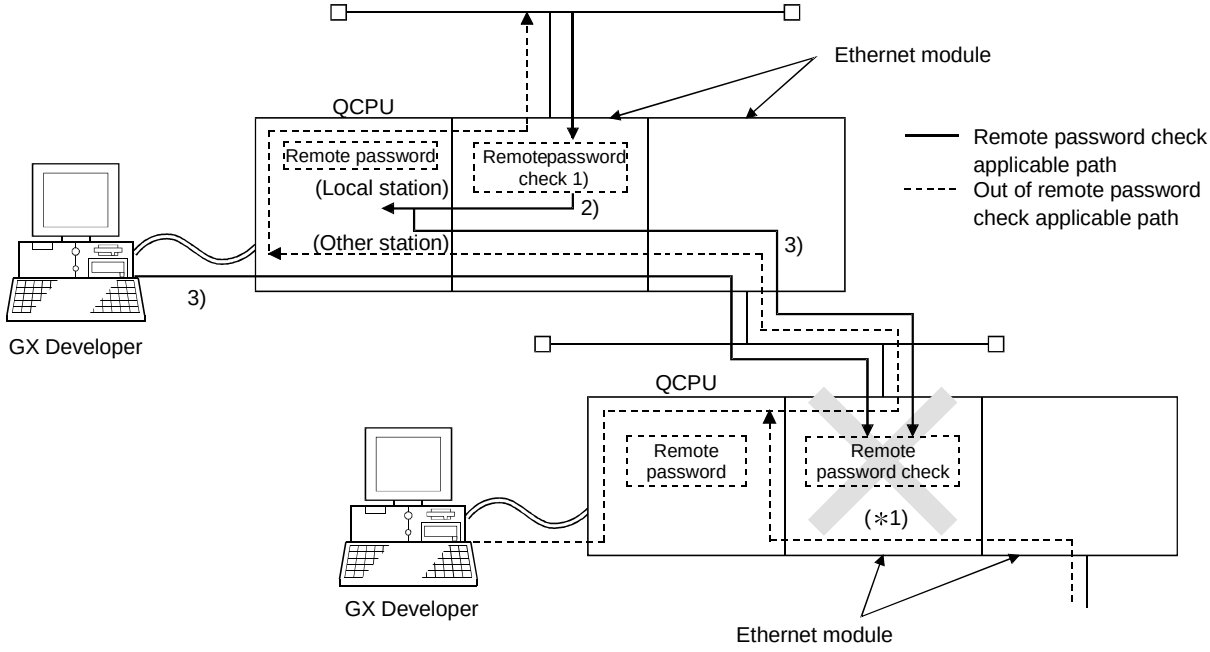
(3) Remote password check performed by the Ethernet module

(a) Communication in which a remote password check is performed

1) When the following parameters have been set for the Ethernet module mounted on the QCPU station, the Ethernet module performs a remote password check for communication requests as indicated below:
- When a remote password is set in the QCPU.
- When the connection that is communicating data with the external device is set as a target for the remote password check.

2) The Ethernet module performs a remote password check with respect to a communication request to the local station/other station received from the external device.

3) The Ethernet module performs transmission with respect to the following transmission requests without performing a remote password check:
- Transmission request from the local station QCPU (such as transmission using the fixed buffer).
- Communication request from the external device (including GX Developer connected to the local station QCPU) to send to another station upon request from the QCPU.



∗1 In the above diagram, a communication request from the external device cannot be received since a remote password check has been set.
If the remote password check has not been set, a communication request can be received and data communication from the external device is possible.

(b) Selecting a connection for which the remote password check is performed
The user can select any connection for which the Ethernet module performs a remote password check and set this using QCPU parameters. (This is set on the "Setting the remote password" screen of GX Developer.)

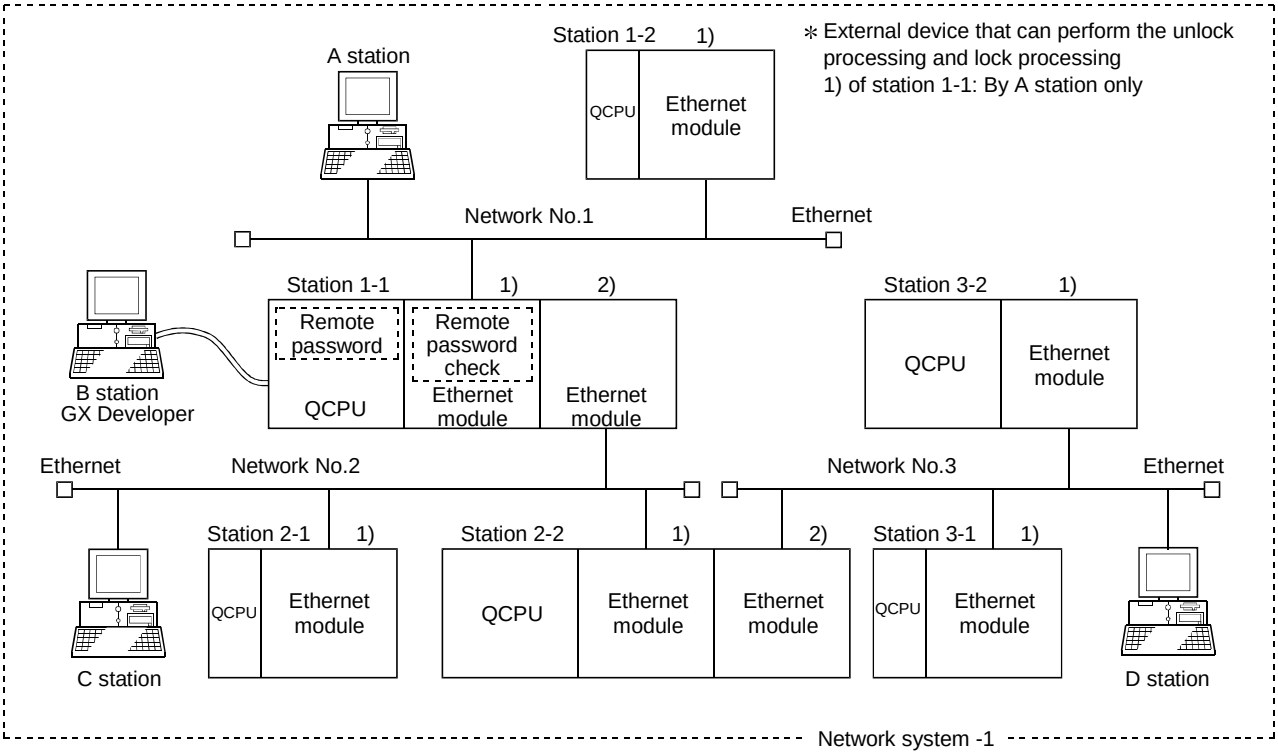
1) User connections (connections 1 to 16)

2) System connections (such as GX Developer communication port)

(c) Stations that can be accessed when the remote password check is performed
The stations that can be accessed from the external device when a remote password is set in the QCPU and the QCPU stations for which the remote password unlock processing and lock processing can be performed are indicated.

(Example 1)

When a remote password is set in the QCPU station of the PLC system station 1-1, and the remote password check is set in 1) of station 1-1



(Request destination) / (Request source)		Applicable PLC station					
		Station 1-1 QCPU	Station 1-2 QCPU	Station 2-1 QCPU	Station 2-2 QCPU	Station 3-1 QCPU	Station 3-2 QCPU
External device (∗2)	A station	●	○	●	●	●	●
	B station	○	○	○	○	○	○
	C station	○	○	○	○	○	○
	D station	○	○	○	○	○	○

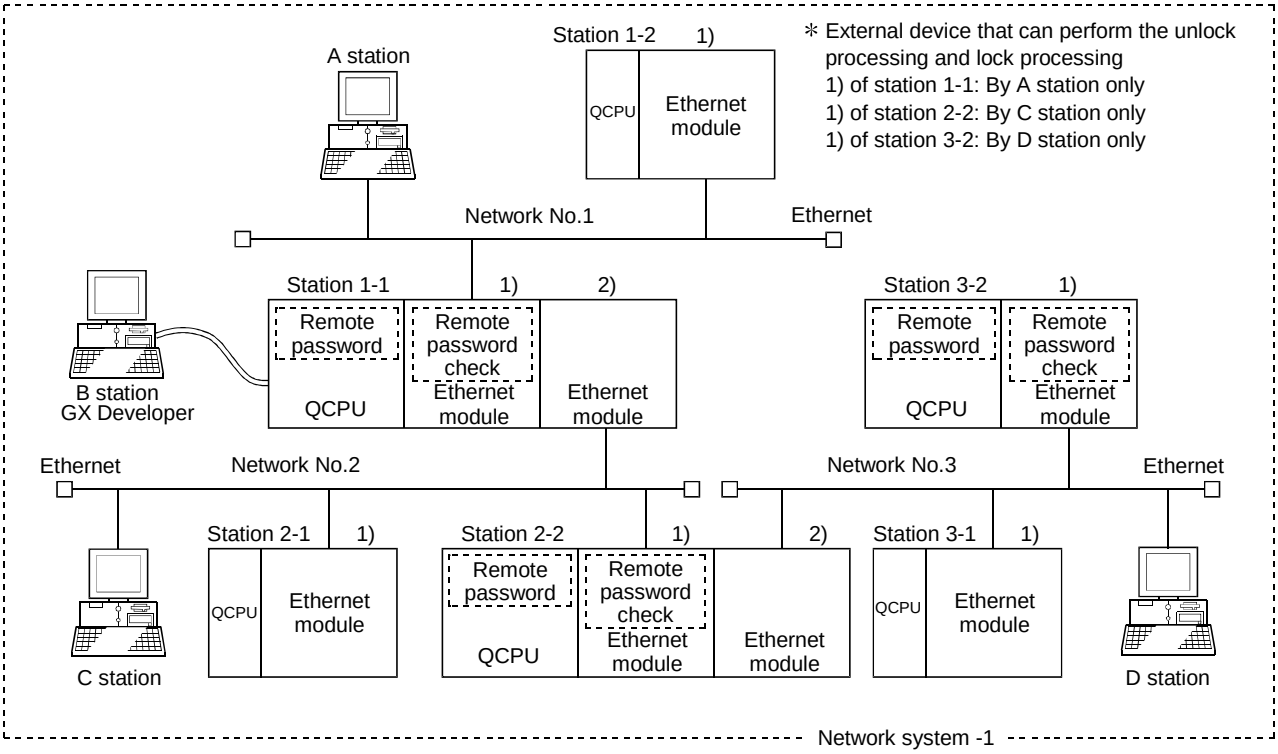
●: Stations that can be accessed form the external device after the remote password unlock processing

○: Stations that can be accessed from the external device without performing the remote password unlock processing

∗2 A station can access the ● stations after the remote password unlock processing for the module 1) of station 1-1 is completed. The ○ stations can be accessed if the communication line is open. B station, C station and D station can access the ○ stations if the communication line to those stations is open.

5 - 73

(Example 2)
When a remote password and the remote password check are set in multiple QCPU stations in the PLC system



(Request destination) / (Request source)		Applicable PLC station					
		Station 1-1 QCPU	Station 1-2 QCPU	Station 2-1 QCPU	Station 2-2 QCPU	Station 3-1 QCPU	Station 3-2 QCPU
External device (*3)	A station	●	○	●	×	×	×
	B station	○	○	○	×	×	×
	C station	○	○	○	●	●	×
	D station	○	○	○	○	○	●

●: Stations that can be accessed form the external device after the remote password unlock processing
○: Stations that can be accessed from the external device without performing the remote password unlock processing
×: Stations that cannot be accessed from the external device

*3 A station can access the ● stations after the remote password unlock processing for the module 1) of station 1-1 is completed. The ○ stations can be accessed if the communication line is open.
B station can access the ○ stations if the communication line to those stations is open.
C station can access the ● stations after the remote password unlock processing for the module 1) of station 2-2 is completed. The ○ stations can be accessed if the communication line is open.
D station can access the ● stations after the remote password unlock processing for the module 1) of station 3-2 is completed. The ○ stations can be accessed if the communication line is open.

5 - 74

POINT

To prohibit access to another station from the external device using the Ethernet module MELSECNET/H, MELSECNET/10 relay communication function, place a check mark at the following setting items in the remote password setting for the relay station and station to be accessed.

"GX Developer communication port (UDP/IP) (*), dedicated instructions, MELSECNET/H, MELSECNET/10 relay communication port"

* Set this on the remote password setting screen of GX Developer.
* Other stations can be accessed if a check mark is not placed in the above setting items.

5 - 75

5.9.2 Precautions when using the remote password check function

The following are precautions when using the remote password check function for the Ethernet module.

(1) After a remote password is set in the QCPU, reboot the QCPU (PLC No.1 for a multiple CPU system) (reset using the RESET/L.CLR switch or power reset). By rebooting the QCPU, the remote password becomes valid.

(2) Set the remote password check to only the connection to be used in data communication with the external device that can perform the unlock processing and lock processing.
(Example) The remote password check should not be set in the reception connection for receiving data transmitted from the MELSEC PLC CPU when communicating by fixed buffer.
* A check mark should not be placed at the applicable connection on the "Remote password detail settings" screen shown in Section 5.9.5.

(3) The remote password check should not be set, since the remote password check is not performed for the connection performing communication using a non-procedure fixed buffer.

(4) When the external device accesses the PLC other station via the Ethernet module, it may not be able to access the PLC if a remote password has been set in the QCPU of the relay station or station to be accessed. (See Section 5.9.1 (1) and (3).)

(5) Precautions when performing UDP/IP communication

1) Decide the external device that will be communicating.
Never perform data communication with a non-specified external device.

2) Use the Ethernet module existence confirmation function.
Also, always perform the remote password lock processing when completing data communication.
* If the lock processing is not performed, data communication from other devices is allowed until a time out is generated by the Ethernet module existence confirmation function.
For this reason, always specify the following as to the settings from GX Developer for the applicable connection:
  - When performing initial settings, set the start interval timer value and the interval timer value for the existence confirmation function as small as possible.
  - Regarding the open settings, select "Confirm" for the "Existence confirmation" item.
* The existence confirmation is automatically performed when the connection for data communication using the automatic open UDP port is set as a target for the remote password check.

(6) As much as possible, use TCP/IP communication to perform communication from GX Developer using an Ethernet connection.

5.9.3 Data communication procedure

This section explains the procedure when the external device performs data communication using the connection for which a remote password check is performed.

(1) Set the target connection for the remote password check in the "Remote password detail settings" through GX Developer and write the parameters in the QCPU. (See Section 5.9.5.)

(2) After booting the QCPU, write the setting values to the following buffer memory in the Ethernet module as necessary: (See Section 5.9.6 (3).)
Accumulated count designation area for remote password mismatch notification
· · · Address 20592 to 20593 (5070_H to 5071_H)

(3) The remote password unlock (release) processing is performed after the open processing for the connection is completed.
All data received before the unlock processing will be processed as an error. (See the troubleshooting section for error codes.)

(4) When the unlock processing for the remote password is performed from the external device, the specified station can be accessed from the external device if the remote password specified by the user matches the remote password set in the local station QCPU.

(5) After data communication is completed, close the applicable connection after performing the remote password lock processing from the external device.

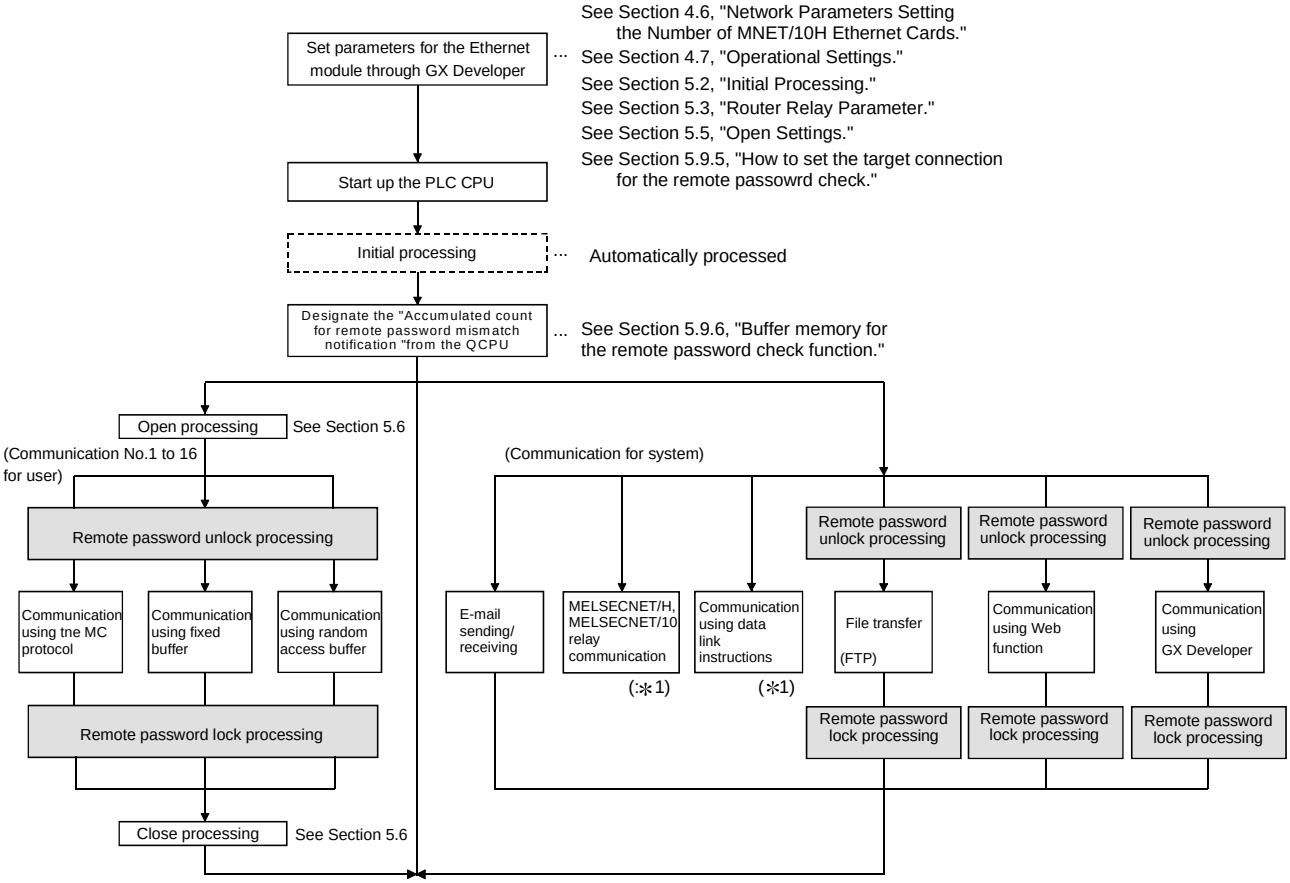
(6) The remote password unlock processing and lock processing are performed from the external device connected to the Ethernet module using the dedicated instructions for MC protocol communication. (They are performed with dedicated FTP commands when using the file transfer (FTP server) function. The remote password is input with the dialog box when using the Web function or GX Developer.)

REMARK

When the "GX Developer communication port" is set as the remote password check connection, access the PLC after performing the unlock processing on the GX Developer screen shown when access begins.

(7) The remote password unlock and lock processing can only be performed for the local station QCPU on which the Ethernet module is mounted. The remote password unlock and lock processing cannot be performed for other station QCPU. (See Section 5.9.1 (2).)

(Data communication procedure)


Set parameters for the Ethernet module through GX Developer
See Section 4.6, "Network Parameters Setting the Number of MNET/10H Ethernet Cards."
See Section 4.7, "Operational Settings."
See Section 5.2, "Initial Processing."
See Section 5.3, "Router Relay Parameter."
See Section 5.5, "Open Settings."
See Section 5.9.5, "How to set the target connection for the remote passowrd check."
Start up the PLC CPU
Initial processing
Automatically processed
Designate the "Accumulated count for remote password mismatch notification "from the QCPU
See Section 5.9.6, "Buffer memory for the remote password check function."
Open processing
See Section 5.6
(Communication No.1 to 16 for user)
(Communication for system)
Remote password unlock processing
Communication using tne MC protocol
Communication using fixed buffer
Communication using random access buffer
E-mail sending/ receiving
MELSECNET/H, MELSECNET/10 relay communication
(*1)
Communication using data link instructions
(*1)
Remote password unlock processing
File transfer (FTP)
Remote password unlock processing
Communication using Web function
Remote password unlock processing
Communication using GX Developer
Remote password lock processing
Remote password lock processing
Remote password lock processing
Remote password lock processing
Close processing
See Section 5.6

*1 When accessing the PLC of other station, access to the other station may not be allowed if a remote password is set in the QCPU of the relay station on which the Ethernet module is mounted and the station to be accessed. (See Section 5.9.1 (1) and (3).)

POINT

(1) In TCP/IP communication, if the connection is closed without performing the remote password lock processing, the Ethernet module automatically performs the lock processing when the connection is closed.

(2) In UDP/IP communication, be sure to close the connection after performing the remote password lock processing.
When the connection is closed, the Ethernet module does not perform the remote password lock processing.

5.9.4 When the remote password unlock processing or lock processing is completed abnormally

This section explains the procedures to be performed by the user when the remote password unlock processing or lock processing is completed abnormally.

(1) Perform the unlock/lock processing again after checking the remote password set in the QCPU.

(2) The Ethernet module performs the following when the number of occurrence times for an unlock processing/lock processing abnormal completion is greater than the accumulated notification count (*1) set in the buffer memory.

(a) The COM.ERR LED lights up.

(b) The code C200H is stored in the error code and end code storage area of the buffer memory error log area (address 227 to 372 (E3H to 174H)).

*1 This is the count set from the QCPU in the remote password mismatch notification accumulated count designation area (address 20592 (5070H), 20593 (5071H)) when the Ethernet module starts up. (Count is set with T0 instruction, etc.)

(3) Monitor the above buffer memory when the COM.ERR LED of the Ethernet module lights up.
If the error code C200H is stored, monitor the storage area for abnormal completion accumulated count in the buffer memory (address 20595 (5073H), 20597 (5075H), · · ·) and check for which connection the unlock processing or lock processing has been completed abnormally.

(4) The user performs the following as required:

(a) Close the applicable connection.

(b) Write "0" in the storage area for abnormal completion accumulated count in the buffer memory.

* If the accumulated count is not cleared by writing "0", the process in (2) above is performed each time an abnormal completion in excess of the accumulated notification count occurs.

(c) If the number of times an unlock processing/lock processing abnormal completion occurred for the applicable connection is greater than the above accumulated notification count, this could indicate improper access from the external device.
Disable the use of the applicable connection using the system port use prohibited designation area of the buffer memory (address 20488 (5008H)). (After this, the unlock processing cannot be performed with respect to applicable connection until the "use allowed" is set.)

(d) Inform the system manager that the number of occurrence times for an unlock processing/lock processing abnormal completion is greater than the accumulated notification count and take appropriate actions.

REMARK

(1) The following accumulated counts stored in the buffer memory can be cleared under the user's option. (Write "0" to the applicable area from the QCPU.)
- Storage area for unlock processing normal completion accumulated count: Address 20594 (5072H) · · ·
- Storage area for lock processing normal completion accumulated count: Address 20596 (5074H) · · ·

(2) See Section 11.1.2 of User's Manual (Basic) for how to turn off the COM.ERR LED for the Ethernet module after it has lit up.

5 - 80

5.9.5 How to set the target connection for the remote password check

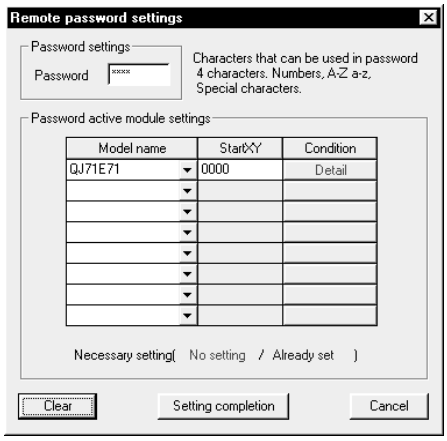
Set the connection for the remote password check with the parameter settings through GX Developer.

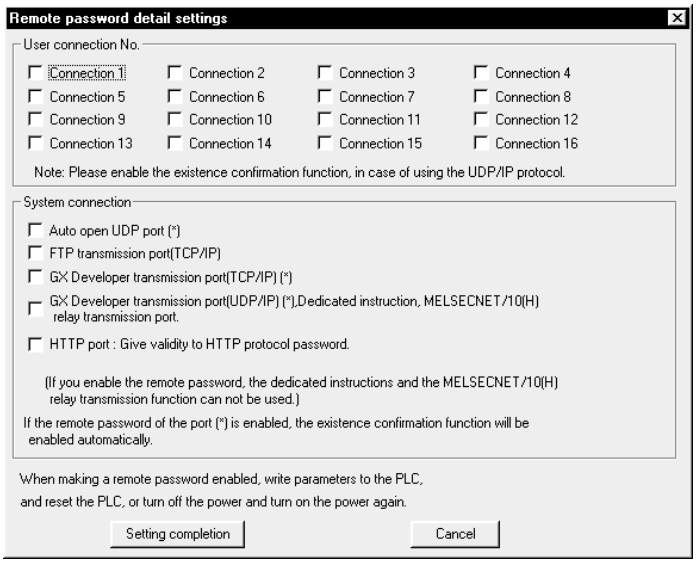
[Startup procedure]

"GX Developer" → Remote password → "Remote password settings" screen

→ Detail → "Remote password detail settings" screen

[Setting screen]




Remote password detail settings
User connection No.
Connection 1
Connection 2
Connection 3
Connection 4
Connection 5
Connection 6
Connection 7
Connection 8
Connection 9
Connection 10
Connection 11
Connection 12
Connection 13
Connection 14
Connection 15
Connection 16
Note: Please enable the existence confirmation function, in case of using the UDP/IP protocol.
System connection
Auto open UDP port (*)
FTP transmission port(TCP/IP)
GX Developer transmission port(TCP/IP) (*)
GX Developer transmission port(UDP/IP) (*),Dedicated instruction, MELSECNET/10(H) relay transmission port.
HTTP port : Give validity to HTTP protocol password.
(If you enable the remote password, the dedicated instructions and the MELSECNET/10(H) relay transmission function can not be used.)
If the remote password of the port (*) is enabled, the existence confirmation function will be enabled automatically.
When making a remote password enabled, write parameters to the PLC,
and reset the PLC, or turn off the power and turn on the power again.
Setting completion
Cancel

[Setting items]

Item name		Description of item setting	Setting range/options
Password settings		Enter the password to be set in the QCPU (*1).	—
Password active module settings	Model name	Select the model name for the module performing the check for the remote password set in the QCPU.	QJ71E71
	Start XY	Set the head address of the module performing the remote password check.	0000H to 0FE0H
	Module condition	(Display the "Remote password detail settings" screen.)	Detail settings
User connection No.		Set the connection subject to the remote password check.	Place a check mark at the target connection.
System connection	Auto open UDP port		
	FTP transmission port (TCP/IP)		
	GX Developer transmission port (TCP/IP)		
	GX Developer transmission port (UCP/IP)		
	HTTP port		

*1 Refer to the following when setting a remote password:

- Avoid using a character string of simple numbers or letters only.
- Combine numbers, letters and special characters (?,!&%,etc.).
- Avoid using a character string that represents the user's name or date of birth.

POINT

(1) When using the Ethernet module in a multiple CPU system, set the remote password by writing it to the control PLC of the Ethernet module.

(2) After the remote password is set in the QCPU, reboot the QCPU (the first QCPU module for a multiple CPU system) (reset using the RESET/L.CLR switch or power reset).
By rebooting the QCPU, the remote password becomes valid.

(3) The password supported by the QCPU of function version A is used to prohibit reading/writing of file data in the QCPU through GX Developer.
Dual access control can be provided by using the remote password described in this section and password for file access.

5.9.6 Buffer memory for the remote password check function

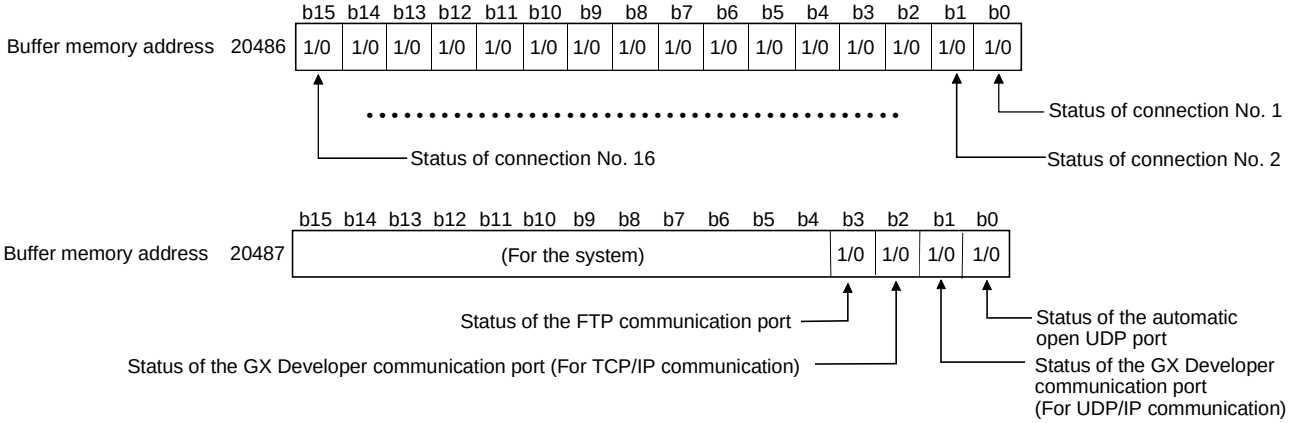
This section explains the buffer memory used for the remote password check function.

Address	Buffer memory for the remote password check function		Application	
			Setting	Monitor
20486 (5006H)	Remote password status (user port)			○
20487 (5007H)	Remote password status (system port)			○
20488 (5008H)	System port use prohibited designation		○	
20592 (5070H)	Remote password mismatch notification accumulated count designation		○	
20593 (5071H)	Remote password mismatch notification accumulated count designation		○	
20594 (5072H)	Connection No. 1	Accumulated count unlock process normal completion		○
20595 (5073H)		Accumulated count unlock process abnormal completion		○
20596 (5074H)		Accumulated count lock process normal completion		○
20597 (5075H)		Accumulated count lock process abnormal completion		○
20598 (5076H)		Accumulated count of lock process based on close		○
:	:	:		
20669 (50BDH)	Connection No. 16	Accumulated count unlock process normal completion		○
20670 (50BEH)		Accumulated count unlock process abnormal completion		○
20671 (50BFH)		Accumulated count lock process normal completion		○
20672 (50C0H)		Accumulated count lock process abnormal completion		○
20673 (50C1H)		Accumulated count of lock process based on close		○
20674 (50C2H)	Connection for automatic open UDP port	Accumulated count unlock process normal completion		○
20675 (50C3H)		Accumulated count unlock process abnormal completion		○
20676 (50C4H)		Accumulated count lock process normal completion		○
20677 (50C5H)		Accumulated count lock process abnormal completion		○
20678 (50C6H)		Accumulated count of lock process based on close		○
20679 (50C7H)	Connection for GX Developer communication (for UDP/IP communication)	Accumulated count unlock process normal completion		○
20680 (50C8H)		Accumulated count unlock process abnormal completion		○
20681 (50C9H)		Accumulated count lock process normal completion		○
20682 (50CAH)		Accumulated count lock process abnormal completion		○
20683 (50CBH)		Accumulated count of lock process based on close		○
20684 (50CCH)	Connection for GX Developer communication (for TCP/IP communication)	Accumulated count unlock process normal completion		○
20685 (50CDH)		Accumulated count unlock process abnormal completion		○
20686 (50CEH)		Accumulated count lock process normal completion		○
20687 (50CFH)		Accumulated count lock process abnormal completion		○
20688 (50D0H)		Accumulated count of lock process based on close		○
20689 (50D1H)	Connection for FTP communication	Accumulated count unlock process normal completion		○
20690 (50D2H)		Accumulated count unlock process abnormal completion		○
20691 (50D3H)		Accumulated count lock process normal completion		○
20692 (50D4H)		Accumulated count lock process abnormal completion		○
20693 (50D5H)		Accumulated count of lock process based on close		○

(1) Remote password status storage area (address 20486 to 20487 (5006H to 5007H))

(a) The current unlock/lock status of the remote password for each connection is stored.
The unlock/lock status for connection No. 1 to No. 16 that use the user port is stored in the address 20486.
The unlock/lock status for connections that use the system port is stored in the address 20487.

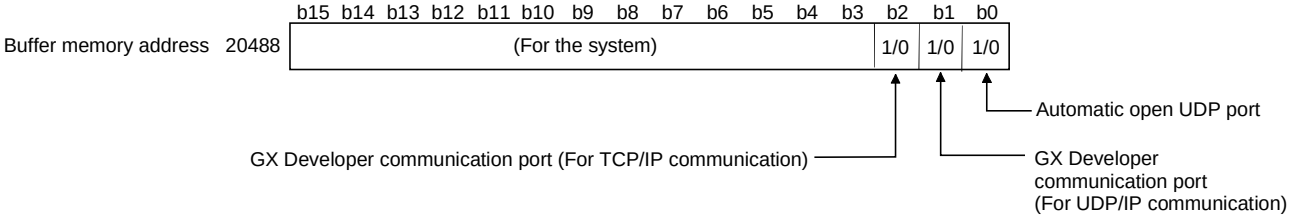
(b) The remote password status for each connection is indicated in bits as follows:
1 (ON): Lock status
0 (OFF): Unlock status/no remote password check setting


b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Buffer memory address 20486
1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0 1/0
Status of connection No. 16
Status of connection No. 1
Status of connection No. 2
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Buffer memory address 20487
(For the system)
1/0 1/0 1/0 1/0
Status of the FTP communication port
Status of the GX Developer communication port (For TCP/IP communication)
Status of the automatic open UDP port
Status of the GX Developer communication port (For UDP/IP communication)

(2) System port use prohibited designation area (address 20488 (5008H))

(a) The opposite device is allowed/prohibited from performing the following data communication using the port provided by the Ethernet module system:
- Communication using the automatic open UDP port (communication using the MC protocol)
- Communication from GX Developer using the Ethernet connection

(b) This designation is determined by the user.
1 (ON): Use prohibited
0 (OFF): Use allowed (default)


b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Buffer memory address 20488
(For the system)
1/0 1/0 1/0
Automatic open UDP port
GX Developer communication port (For TCP/IP communication)
GX Developer communication port (For UDP/IP communication)

(c) If the remote password check is performed and the remote password status of the connection using the applicable port is in the lock status, the use of the applicable port can be prohibited.

(d) If improper access from the opposite device has been performed during the above data communication, set the applicable port to "use prohibited." (See Section 5.9.4 for details.)

REMARK

To prohibit data communication from the opposite device using the Ethernet module FTP function, set "Not used" in the FTP function setting item under the "Setting the Ethernet FTP parameters" through GX Developer. (See Section 5.3 of User's Manual (Application).)

(3) Remote password mismatch notification accumulated count designation area (Address 20592 to 20593 (5070H to 5071H))

(a) Use 0 to FFFFH to designate the count that will be the notification timing to the user when the number of occurrence of remote password mismatches become the permit or more during the unlock/lock processing from the user/external device after the Ethernet module starts up. (This designation is common to all connections.)
0H: No designation
(The process (c) when a remote password mismatch occurs is not performed.)
1 to FFFFH: Notification accumulated count

(b) Designate the notification accumulated count for each of the following connections in this area:
Address 20592 area: Connection No. 1 to No. 16
Address 20593 area: Connection using the automatic open UDP port
Communication from GX Developer using the Ethernet connection
Communication from the external device that uses the FTP function

(c) When the number of times a remote password mismatch occurred during the unlock/lock processing exceeds the notification accumulated count, the Ethernet module performs the following a mismatch occurs. (See Section 5.9.4 for details.)
- The COM.ERR LED lights up.
- The code C200H is stored in the error code and end code storage area of the buffer memory error log area (address 227 to 372 (E3H to 174H)).

＊ The Ethernet module system does not close the connection.

(d) The accumulated number of times a remote password mismatch occurred up to the present (accumulated count value according to the Ethernet module) can be checked using the storage area for abnormal completion accumulated count indicated in (4) below.

(4) Storage area for accumulated count unlock process normal completion of (address 20594 (5072H) · · ·)
Storage area for accumulated count unlock process abnormal completion of (address 20595 (5073H) · · ·)
Storage area for accumulated count lock process normal completion of (address 20596 (5074H) · · ·)
Storage area for accumulated count lock process abnormal completion of (address 20597 (5075H) · · ·)

(a) The accumulated number of times the remote password unlock/lock processing for the applicable connection has been completed normally/abnormally up to present is stored.

(b) The user should clear the value stored in each of the accumulated count storage areas. (Write "0" to the applicable area from the QCPU.)

(5) Storage area for accumulated count of lock process based on close (Address 20598 (5076H) · · ·)

(a) When the user closes the applicable connection without performing the remote password lock processing, the Ethernet module will automatically perform the lock processing.
The accumulated number of times the Ethernet module performed the lock processing automatically is stored in this area.

(b) The user should clear the value stored in this accumulated count storage area. (Write "0" to the applicable area from the QCPU.)

POINT
The maximum value that can be stored in the areas indicated in (4) and (5) above is FFFFH. An accumulated count that exceeds FFFFH (65535) is not stored.

5 - 86

5.9.7 Data communication when the remote password check is set

The following describes data communication that uses a connection for which the remote password check is set.

<table>
<tr><th colspan="2" rowspan="2">Function</th><th colspan="2">Data communication</th><th rowspan="2">Remarks</th></tr>
<tr><th>No remote password check setting</th><th>Remote password check setting (∗1)</th></tr>
<tr><td rowspan="2">Communication using the MC protocol</td><td>User open port</td><td>Communication is possible after the open processing is completed.</td><td>After the open processing is completed, communication is allowed from the time the unlock command is received until the lock command is received.</td><td rowspan="2">See Section 3.18 of the Reference Manual for the unlock and lock commands.</td></tr>
<tr><td>Automatic open UDP port</td><td>Communication is possible after the initial processing is completed.</td><td>After the initial processing is completed, communication is allowed from the time the unlock command is received until the lock command is received.</td></tr>
<tr><td rowspan="3">Communication using the fixed buffer</td><td>There is a procedure</td><td rowspan="3">Communication is possible after the open processing is completed.</td><td>After the open processing is completed, communication is allowed from the time the unlock command is received until the lock command is received.</td><td rowspan="8">—</td></tr>
<tr><td>Non procedure</td><td>Communication is allowed after the open processing is completed. (∗2)
∗ Since the remote password check is not performed in non procedure communication, the remote password check is not set for the connection used.</td></tr>
<tr><td colspan="2">Communication using the random access buffer</td><td>After the open processing is completed, communication is allowed from the time the unlock command is received until the lock command is received.</td></tr>
<tr><td colspan="2">Sending/receiving of e-mail</td><td colspan="2">After the initial processing is completed, transmission and reception are allowed.
∗ The remote password check is not performed for e-mail transmission and reception.</td></tr>
<tr><td colspan="2">Communication using Web function</td><td>Communication is possible after the initial processing is completed.</td><td>Communication is allowed after the remote password is entered. The remote password lock processing is automatically performed when the web browser closes.</td></tr>
<tr><td colspan="2">MELSECNET/H, MELSECNET/10 relay communication</td><td rowspan="2">Communication is possible after the initial processing is completed.</td><td rowspan="2">Communication is possible after the initial processing is completed. (∗3)</td></tr>
<tr><td colspan="2">Communication using data link instructions</td></tr>
<tr><td colspan="2">File transfer (FTP server) function</td><td>Within the Ethernet, communication is possible with the external device that has completed the open processing.</td><td>After the open processing is completed, communication is allowed from the time the unlock command is received until the lock command is received.</td><td>See Chapter 5 of User's Manual (Application).</td></tr>
<tr><td colspan="2">GX Developer TCP communication (∗1)</td><td rowspan="2">After the initial processing is completed, communication is allowed by establishing a connection from the GX Developer side.</td><td rowspan="2">Communication is allowed after the remote password is entered.
The remote password lock processing is automatically performed when the project closes.</td><td rowspan="2">See the operating manual for GX Developer.</td></tr>
<tr><td colspan="2">GX Developer UDP communication (∗1)</td></tr>
</table>

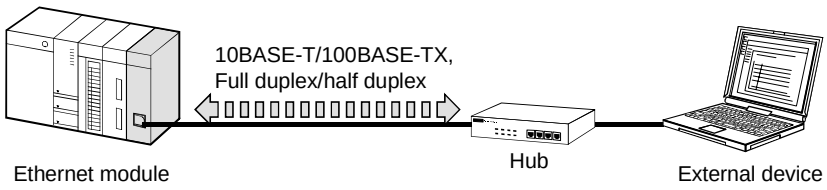
∗1 See Section 5.9.1 (3) for access to the PLC of other stations.

∗2 A dedicated connection is used for the non-procedure fixed buffer communication. Do not perform the remote password setting for the applicable connection.

∗3 If a remote password has been set in the QCPUs of the relay station and access station where Ethernet modules are installed, when accessing other station's PLC, it may not be possible to access other stations. (See items (1) and (3) in Section 5.9.1.)

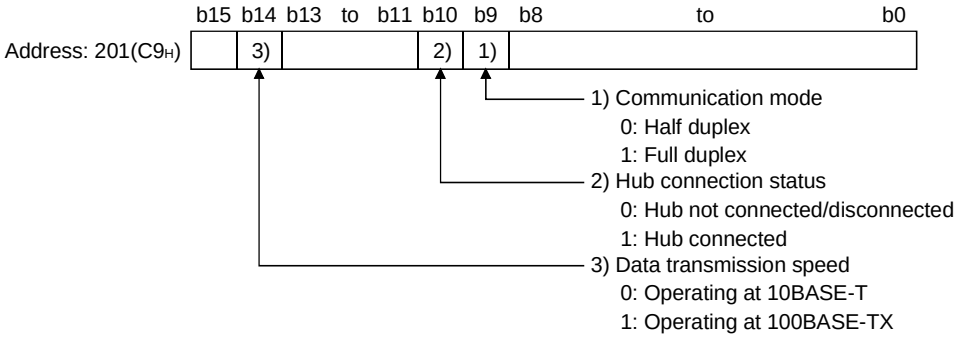
5.10 Hub Connection Status Monitor Function

The current connection status of the Ethernet module and hub, the transmission speed, and the number of times the Ethernet module detected disconnection can be checked at the following buffer memory addresses. (QJ71E71-100 only)



(1) Hub connection status area (Address: 201 (C9H))

Stores the current connection status of the QJ71E71-100 and hub and the transmission speed.



(Bits other than above are reserved for system use.)

(2) Disconnection detection count storage area (Address: 20995 (5203H))

(a) Stores the number of disconnection detection times after initial processing is completed (X19 turns ON).
Disconnection is detected in any of the following cases.
- Disconnection between Ethernet module and hub
- Cable removal from hub side connector
- Hub power-off
- Cable removal from Ethernet module side connector

(b) If an error has occurred 65536 times or more, a count stops at FFFFH (65535).
Write "0" to this area using a sequence program to clear the stored value.

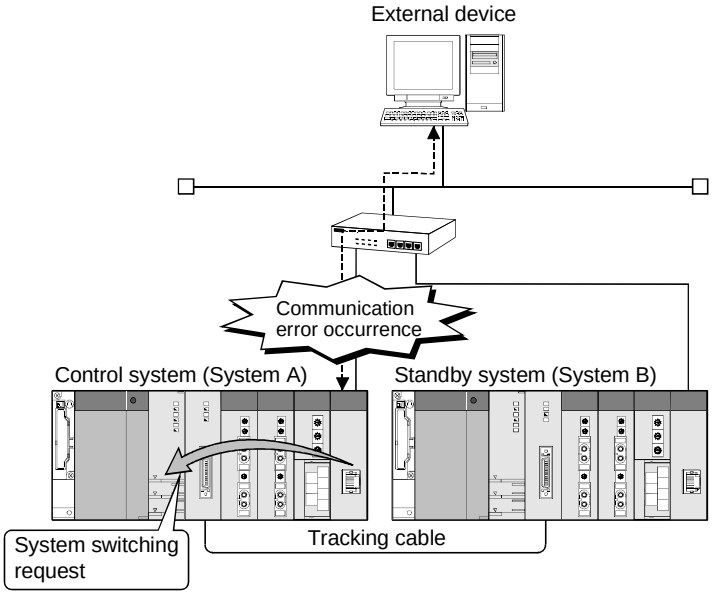
5.11 Configuring a Network in Redundant System (Redundant System Support Function)

This section explains the functions and settings for use of the Ethernet module in the redundant system. Refer to Section 2.5 for the system configuration.

5.11.1 Issue of System Switching Request to Control System CPU

(1) "Issue of System Switching Request to Control System CPU"

This function issues a system switching request to the control system CPU when the Ethernet module mounted on the main base unit of the control system CPU in the redundant system detects a communication error or disconnection.



(2) "System switching request issuing condition"

When any of the following conditions set in the redundant setting of GX Developer is satisfied, the Ethernet module issues a system switching request to the control system CPU. (Refer to Section 5.11.3 for the redundant setting.)

System switching request issuing condition		Description
Communication error detection	Existence check	After the connection is opened (after open processing), the existence of the external device cannot be checked.
	ULP timeout	There is no ACK response from the external device within the TCP ULP timer value.
Disconnection detection		The cable connected to the Ethernet module was disconnected. (QJ71E71-100 only)

POINT

When any of the system switching request issuing conditions indicated in (2) of this section is satisfied, the system is switched between the control system and standby system. In either of the following cases, however, the system will not be switched if the system switching request is issued from the Ethernet module.

- The standby system is already in a faulty status (e.g. power-off, reset or stop error) (For the reasons for system switching, refer to the QnPRHCPU User's Manual (Redundant System).)
- The network module redundant group settings have been mode for Ethernet modules and either one is operating normally (For the "network module redundant group setting", refer to the QnPRHCPU User's Manual (Redundant System).)

(3) Issue of system switching request at "communication error"

The Ethernet module mounted on the main base unit of the control system CPU monitors communication with the external device on each connection. When detecting a communication error is it issues a system switching request to the control system CPU.

(a) "Communication error at which system switching request is issued"

When either of the following communication errors occurs, the system switching request is issued.

Communication error	Description
Existence check	After the connection is opened (after open processing), the existence of the external device cannot be checked.
ULP timeout	There is no ACK response from the external device within the TCP ULP timer value.

(b) "Target connection"

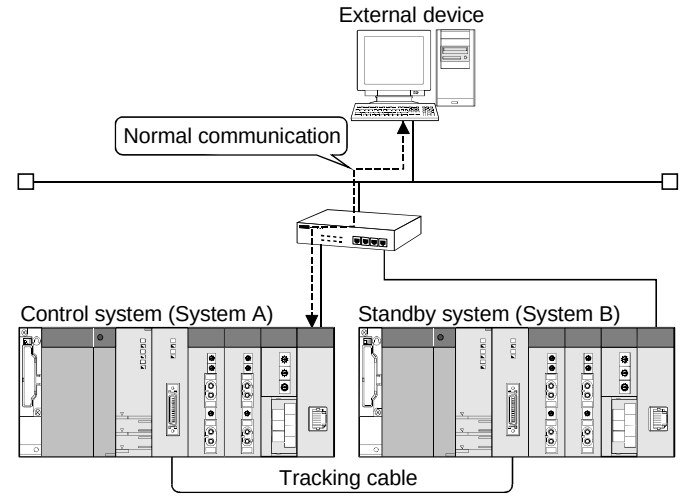
A communication error is detected on the connections set in the "system switching settings when communication error occurs" in the redundant setting of GX Developer. (Refer to Section 5.11.3 for the redundant settings.)

The following connections can be the targets of monitoring.

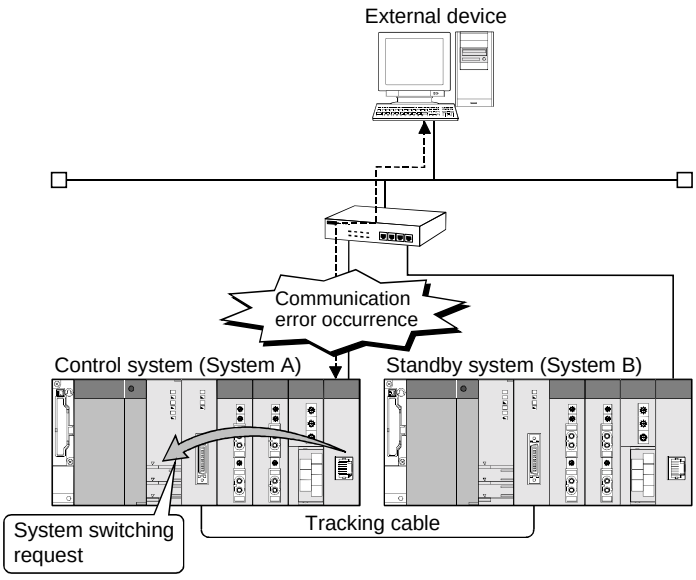
- Connection No. 1 to 16
- Automatic open UDP port
- FTP communication port
- GX Developer communication port (TCP, UDP)
- HTTP communication port

(c) System switching request operation at "communication error"

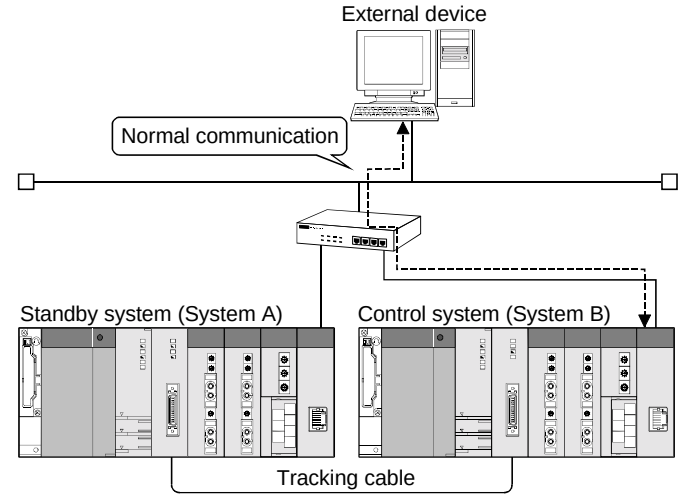
The Ethernet module checks communication with the external device for an error using the existence check function or TCP ULP timer.
The system switching request operation performed at a communication error is described below.



1) During normal communication
System A is operating as a control system, and system B as a standby system.
The external device is communicating with the Ethernet module mounted on the main base unit of the control system CPU. (∗1)



2) At error detection (∗2)
When the Ethernet module mounted on the main base unit of the control system CPU detects the communication error that occurred between the external device and Ethernet module, it issues a system switching request to the control system CPU. (∗3)



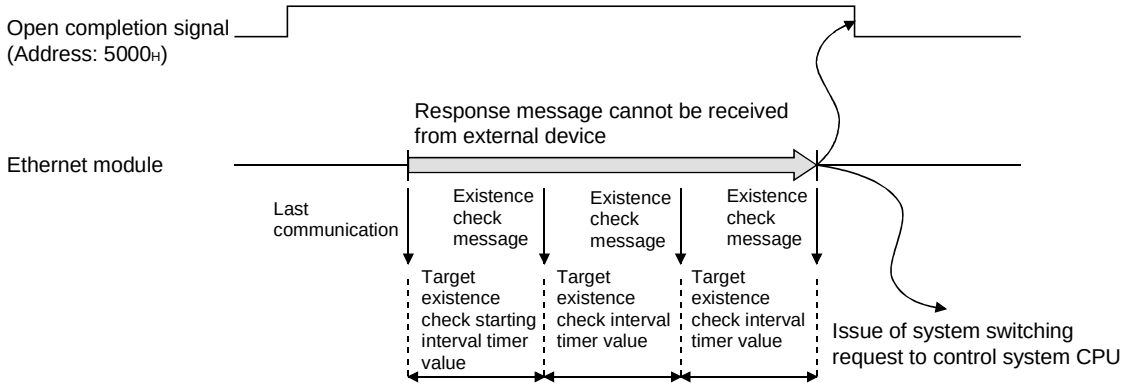
3) After system switching
System A operates as a standby system, and the system B as a control system.
The external device changes the connection destination, and communicates with the Ethernet module mounted on the main base unit of the control system CPU (System B).

5 - 91

∗1 By connecting the external device and the Ethernet module mounted on the main base unit of the standby system CPU via TCP, a standby system CPU side error can be detected.

∗2 System switching timing at communication error
The timing of issuing the system switching request to the control system CPU at a communication error is shown below.

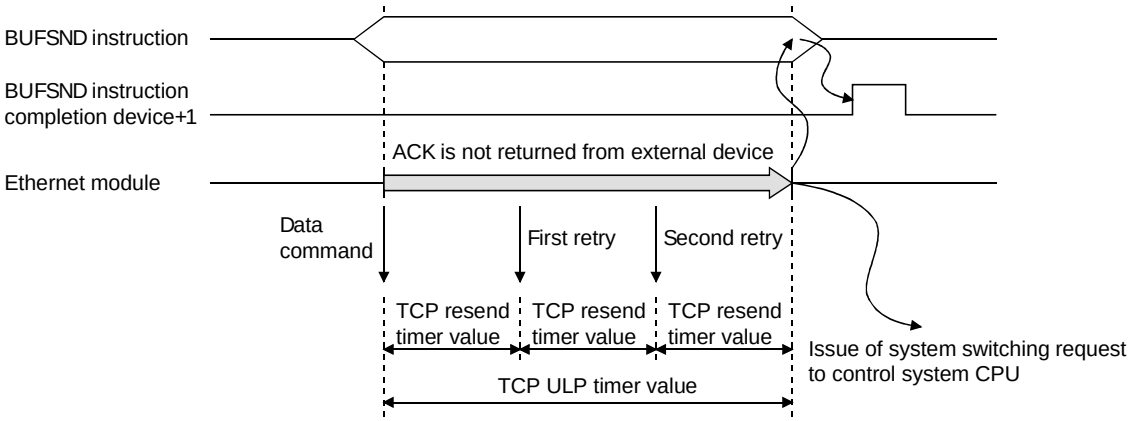
(a) System switching timing by "existence check"



1) The Ethernet module makes an existence check on the external device whose connection is open if communication is not made for a predetermined period of time. (Refer to Section 5.2.2 for the existence check function.)

2) If the Ethernet module cannot receive a response message from the external device, it closes the corresponding connection and issues a system switching request to the control system CPU.
(The above chart shows an existence check example of two resend times.)

(b) System switching timing by "ULP timeout"



1) If ACK is not returned from the external device within the TCP ULP timer period at TCP opening or data transmission, a transmission error occurs and the Ethernet issues a system switching request to the control system CPU. (Refer to Section 5.2.2 for the TCP ULP timer.)
(The above chart shows a setting example of two retries.)

∗3 Set whether the system switching request will be issued or not to the control system CPU in the redundant settings of GX Developer. (Refer to Section 5.11.3.)

(4) Issue of system switching request at "disconnection detection"

The Ethernet module mounted on the main base unit of the control system CPU monitors the connection status of the cable connected to the Ethernet module, and on detection of disconnection, it issues a system switching request to the control system CPU.

This function is available for the QJ71E71-100 only.

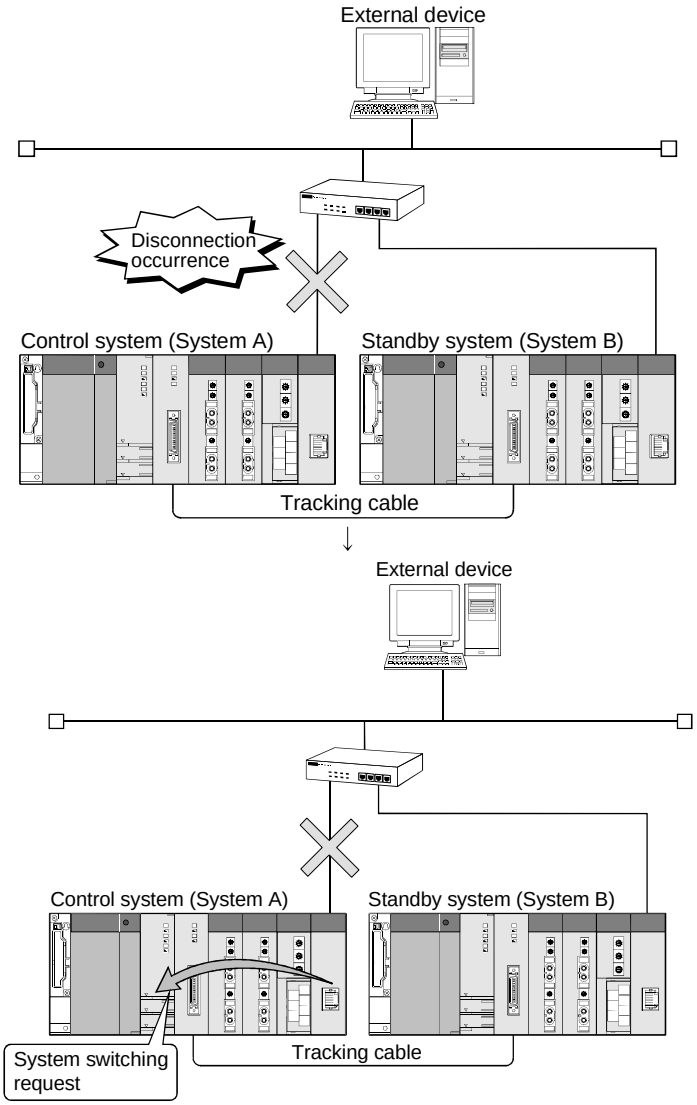
(a) "Disconnection detection at which system switching request is issued"

Disconnection is detected in any of the following cases.

- Disconnection between Ethernet module and hub
- Cable removal from hub side connector
- Hub power-off
- Cable removal from Ethernet module side connector

(b) System switching request operation at "disconnection detection"

The Ethernet module always checks the connected cable for disconnection. The system switching request operation performed at disconnection detection is described below.



1) Disconnection monitoring

The Ethernet module always checks the connected cable for disconnection. (∗1) The monitoring result is stored into the hub connection status area (address: 201 (C9H)) of the buffer memory. (Refer to Section 5.10.)

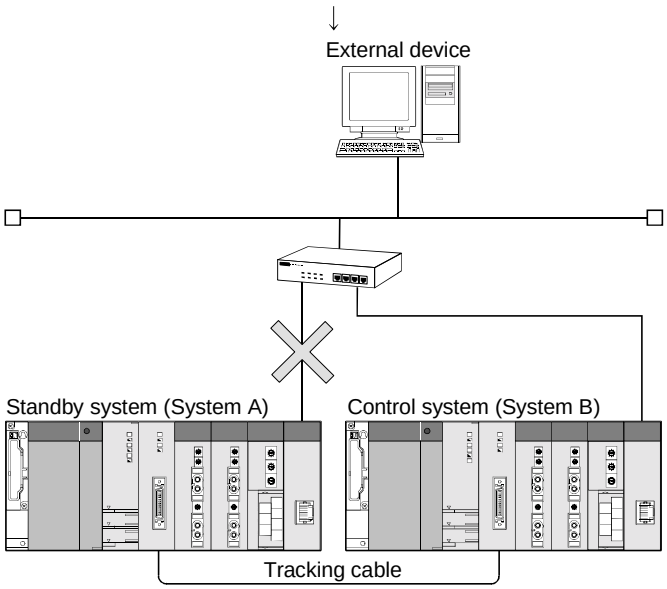
2) At disconnection detection (∗2)

When the Ethernet module mounted on the main base unit of the control system CPU detects disconnection, it executes a disconnection status time check, and when a disconnection status continues for the period of cable disconnection timeout setting, the Ethernet module issues a system switching request to the control system CPU. (∗3)

(To next page)

5 - 93

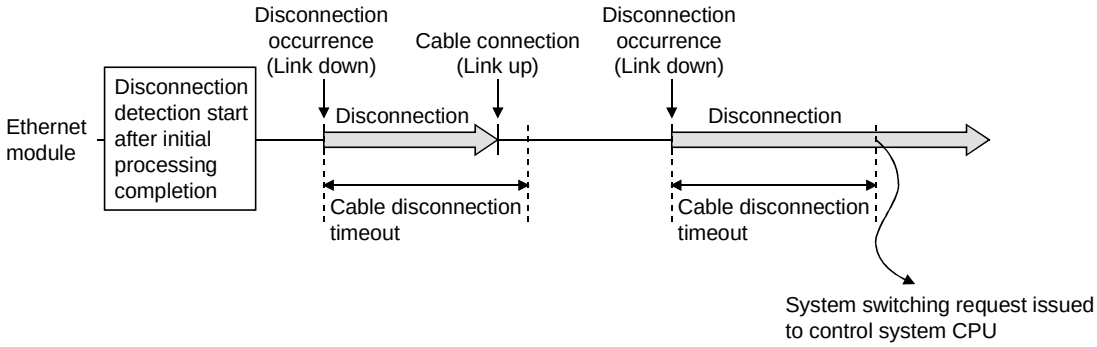
(From previous page)



3) After system switching
System A operates as a standby system, and system B as a control system.

∗1 When the cable is not connected from the start, the Ethernet module does not regard it as disconnection. (Disconnection is detected only when normal status turns to abnormal.)

∗2 System switching timing at disconnection detection
The timing of issuing a system switching request to the control system CPU at disconnection detection is shown below.



1) The Ethernet module starts disconnection detection after initial processing completion.
2) On detection of disconnection, the Ethernet module performs a disconnection status time check, and when a disconnection status continues for the period of cable disconnection timeout setting, the Ethernet module issues a system switching request to the control system CPU.
3) When the disconnection status returns to normal within the cable disconnection timeout setting, the Ethernet module does not issue a system switching request.

∗3 Set whether the system switching request will be issued or not to the control system CPU in the redundant setting of GX Developer. (Refer to Section 5.11.3.)

5.11.2 Communication path bypass function

(1) Communication path bypass function

When any of the following redundant system-support applications is used, the path where a communication error occurred can be automatically bypassed to continue communication if an error occurs in communication with the Ethernet module.

The user need not change from one communication path to the other.

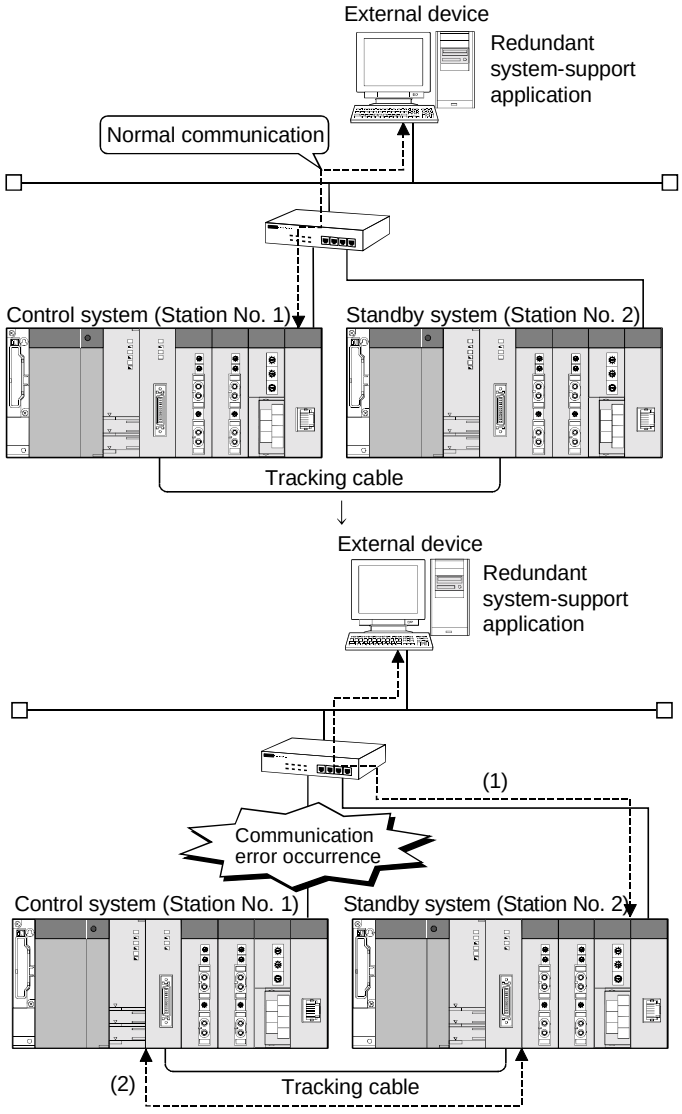
- Application that operates on the OPS (except the MELSOFT products such as GX Developer) ∗1
- GX Developer
- PX Developer monitor tool

For the compatible versions and available functions, refer to the manual of the corresponding application.

∗1 The Ethernet module can communicate with the OPS using the user connection for OPS connection.
Set the user connection for OPS connection in the open settings of GX Developer. (Refer to Section 5.5.)

(2) Example of redundant system-support application operation at occurrence of communication error

The following provides an example of redundant system-support application operation performed when an error occurs during communication with the control system CPU.



(1) Path is bypassed to make access to control system CPU.
(2) Communication with control system CPU is continued using tracking cable.

1) During normal communication
Station No. 1 is operating as a control system, and station No. 2 as a standby system.
The redundant system-support application is communicating with the control system CPU.

2) At communication error occurrence
Station No. 1 is operating as a control system, and station No. 2 as a standby system.
(Without system switching) (*1)

[Operation of redundant system-support application]
Since a communication error occurred between the redundant system-support application and the Ethernet module mounted on the main base unit of the control system CPU, the redundant system-support application changes the communication path automatically and communicates with the control system CPU via the standby system CPU.

[Operation of standby system CPU]
Since the received data is addressed to the control system CPU, data is relayed to the control system CPU via the tracking cable.

*1 Set whether the system switching request will be issued or not to the control system CPU in the redundant settings of GX Developer. (Refer to Section 5.11.3.)

5 - 96

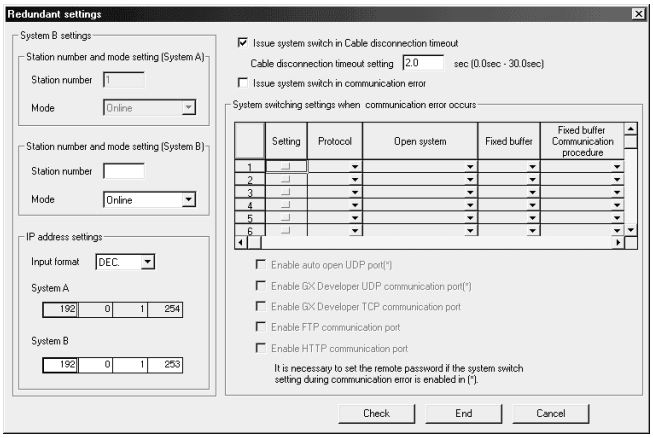
5.11.3 Redundant settings

This section explains the redundant settings of GX Developer.

[Starting procedure]

[Setting the number of MNET/10H Ethernet cards] → Redundant settings → "Redundant settings screen"

[Setting screen]



[Setting items]

Item name	Description
System B settings	Set the station number and IP address of the Ethernet module mounted on the system B.
Issue system switch in Cable disconnection timeout	Set whether a system switching request will be issued or not to the control system CPU at disconnection detection.
Issue system switch in communication error	Set whether a system switching request will be issued or not to the control system CPU at a communication error.

(1) System B settings

Set the station number, mode and IP address of the Ethernet module mounted on the system B.

The setting method is the same as that of the Ethernet module mounted on system A.

Refer to Section 4.6 and Section 4.7.

POINT

(1) Set different station numbers and IP addresses to system A and system B.

(2) When using the redundant system in the backup mode, set the operation mode of system A as that of system B.
If the mode of the Ethernet module differs between system A and system B, an error will occur in the redundant CPU.

(3) For the Ethernet module settings other than the station number, mode and IP address, use the same data for system A and system B.

(4) Set the station number, mode and IP address of the Ethernet module mounted on System A in "Setting the number of MNET/10H Ethernet cards" and "Operational settings".

(2) "Issue system switch in Cable disconnection timeout"

When this setting is made valid, the Ethernet module issues a system switching request to the control system CPU when a disconnection status continues for a period of disconnection monitoring time after detection of the disconnection.
Refer to Section 5.11.1 for system switching at disconnection detection.

Item	Description	Setting range/choices
Issue system switch in Cable disconnection timeout	Set whether a system switching request will be issued or not at disconnection detection.	• Checked (System switching request is issued) • Not checked (System switching request is not issued) (Default: Checked)
Cable disconnection timeout setting	Set the time from when disconnection is detected until the system switching request is issued to the control system CPU.	0.0s to 30.0s (Default: 2.0s)

POINT

Set the cable disconnection timeout setting to several seconds to several ten seconds.
If it is shorter than the above, a system switching request may be generated due to noise, etc.

(3) "Issue system switch in communication error"

When this setting is made valid, the Ethernet module issues a system switching request to the control system CPU when a communication error is detected on the connection set in the "system switching settings when communication error occurs".
Refer to Section 5.11.1 for system switching at communication error.

Item	Description	Setting range/choices
Issue system switch in communication error	Set whether a system switching request will be issued or not at communication error.	• Checked (System switching request is issued) • Not checked (System switching request is not issued) (Default: Not checked)
System switching settings when communication error occurs ∗1∗2∗3	Set the target connection whose communication error will cause a system switching request to be issued.	Check the target connection. (Default: Not checked)

∗1 The settings of the automatic open UDP port and GX Developer communication UDP port are made valid when the following conditions are satisfied.
1) The remote password setting is valid.
2) The remote password is canceled.
When the above conditions are not satisfied, a system switching request is not issued if a communication error occurs on the target connection.

∗2 Do not set the following connection as the target connection. (Do not check its check box.)
• Initial timing setting of operation setting : "Always waiting for OPEN (communication enabled during STOP)"
Protocol of open setting : "UDP"
Existence check of open setting : "Check"
If the above connection is set as the target connection, system switching may occur consecutively in the redundant system when a communication error occurs due to cable disconnection or external device power-off.

∗3 It is recommended not to set the connection, which has been set to "MELSOFT connection" in the open system of the open setting, as the target connection. (Do not check the check box.)
If the above connection is set as the target connection, all the MELSOFT products connected to the network will be targets, and therefore, the target external device (MELSOFT product) cannot be identified.

5.11.4 Buffer memory for redundant system support function

The redundant setting status of GX Developer can be checked in the following buffer memory areas.

(1) "Issue system switch in Cable disconnection timeout" (Address: 20992 (5200H))

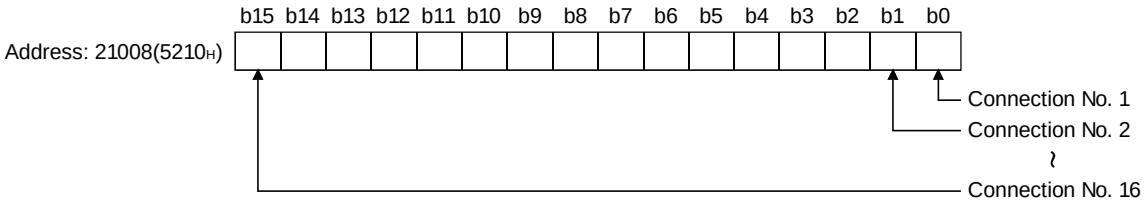
Stores the setting status of "Issue system switch in Cable disconnection timeout".
0: Not set
1: Set

(2) Disconnection timeout setting (Address: 20993 (5201H))

Stores the setting status of the "disconnection timeout setting".
Set time = set value × 500ms
(Example) When the set time is 2s, the storage value is 4H.

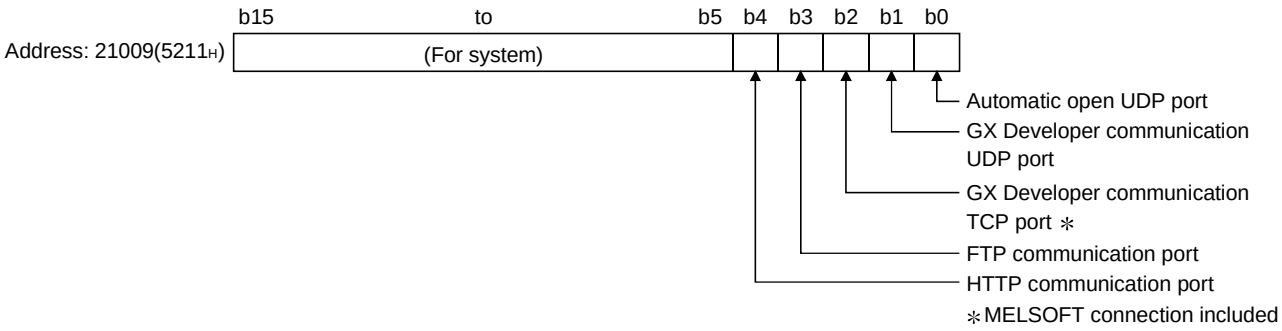
(3) "System switching settings when communication error occurs" (Connection for user) (Address: 21008 (5210H))

Stores the setting status of "System switching settings when communication error occurs" to the connection for user.
0: Not set
1: Set



(4) "System switching settings when communication error occurs" (Connection for system) (Address: 21009 (5211H))

Stores the setting status of "System switching settings when communication error occurs" to the connection for system.
0: Not set
1: Set


b15 to b5 b4 b3 b2 b1 b0
Address: 21009(5211H)
(For system)
Automatic open UDP port
GX Developer communication UDP port
GX Developer communication TCP port ∗
FTP communication port
HTTP communication port
∗MELSOFT connection included

5 - 99

5.11.5 Data communication for using Ethernet module in redundant system

This section describes data communication made when the Ethernet module is mounted on the main base unit of a redundant system.
For other than the following, data communication can be made in the same manner as when the Ethernet module is mounted on the main base unit of a single CPU system. Refer to the explanation item of the corresponding function.

(1) "Initial processing"

(a) Execution of initial processing

Make settings for data communication using GX Developer, write the same parameters to the control system CPU and standby system CPU, and then simultaneously reset the both redundant CPUs.
Note that different station numbers and IP addresses must be set to system A and system B. (Refer to Section 5.11.3.)

(b) When performing re-initial processing (Refer to Section 5.2.3)

Do not change the local station IP address and operation settings.
If they are changed, normal communication cannot be made.

1) When using the UINI instruction

Execute the instruction after specifying "0_H" for the change target ((S1+2) of the control data.

2) When directly writing to buffer memory

Write "1" to bit 15 of the communication condition setting area (address: $1F_H$) without changing the set value of the buffer memory area.

(c) Initial processing by I/O signal

Since the output signal (Y) turns off in the standby system CPU, initial processing by I/O signal is unavailable.
Set the network parameters of GX Developer and perform initial processing. (Refer to Section 5.2.)

(2) "Open/close processing"

(a) When making communication via TCP/IP

Place the Ethernet module in an open waiting status (Passive open), and perform open/close processing from the external device.
When Active open processing is performed from the Ethernet module, close processing is performed from the Ethernet module, but if system switching occurs before execution of close processing, close processing cannot be executed.

(b) When using user connection to communicate with standby system (For communication using MC protocol or random access buffer)

1) Operation setting (Refer to Section 4.7)
Set the initial timing setting to "Always waiting for OPEN (communication enabled during STOP)".

2) Open setting (Refer to Section 5.5)
When making TCP/IP communication, set the open system to "Unpassive or Fullpassive".

(c) Open/close processing by I/O signal
Since the output signal (Y) turns off in the standby system CPU, open/close processing by I/O signal is unavailable.
Set the initial timing setting of the operation setting to "Always waiting for OPEN (communication enabled during STOP)", or use the dedicated instruction (OPEN/CLOSE instruction). (Refer to Section 5.6.)

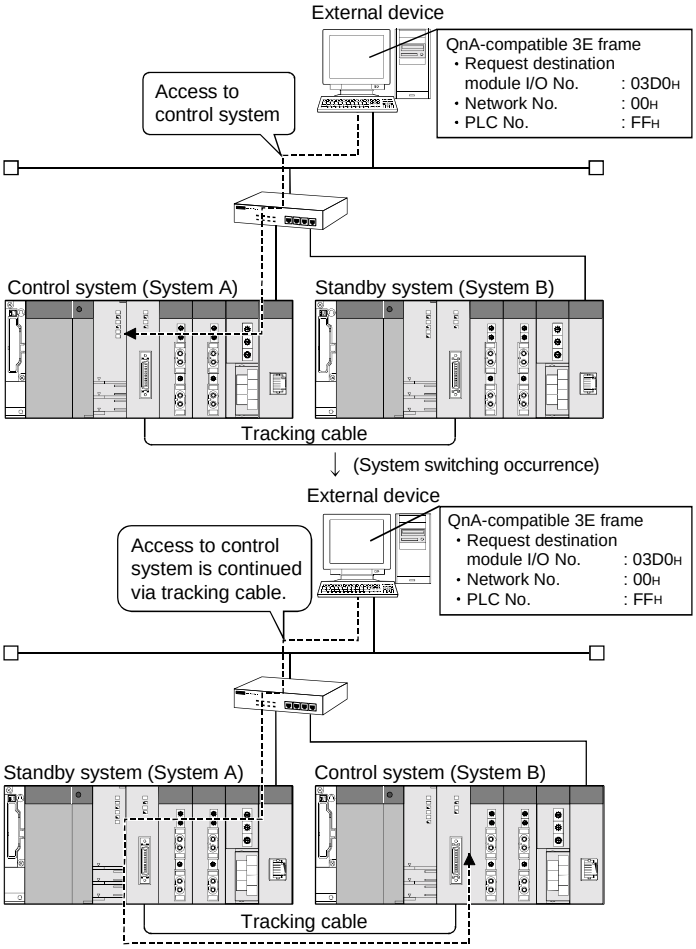
POINT
When using the user connection for communication, it is recommended to prepare the connections for communication with system A and for communication with system B. If a communication error occurs in the host system or the system switching occurs, the above enables immediate communication with the other system. On the Ethernet module, up to 16 user connections can be set.

(3) "Communication using MC protocol"

The QnA-compatible 3E frame can be used to access the control system/standby system or system A/system B.
For the message format and data specification method, refer to the Reference Manual.

(a) Operation performed for access to control system/standby system or system A/system B

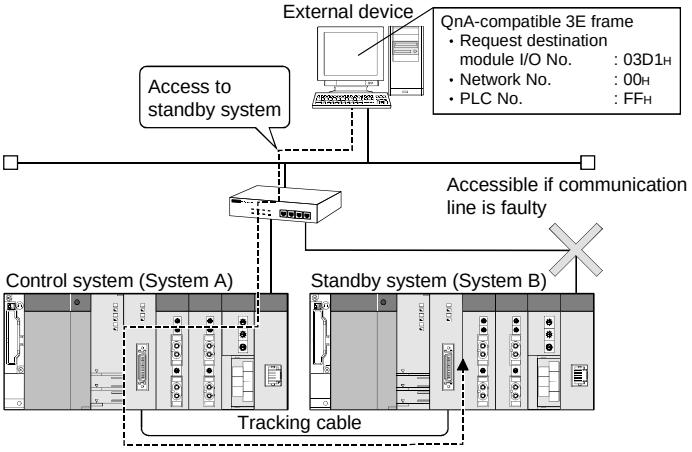
1) When system switching occurred (Example of access to control system CPU)


External device
QnA-compatible 3E frame
・Request destination module I/O No. : 03D0H
・Network No. : 00H
・PLC No. : FFH
Access to control system
Control system (System A)
Standby system (System B)
Tracking cable
(System switching occurrence)
External device
QnA-compatible 3E frame
・Request destination module I/O No. : 03D0H
・Network No. : 00H
・PLC No. : FFH
Access to control system is continued via tracking cable.
Standby system (System A)
Control system (System B)
Tracking cable

1) The external device connects to the Ethernet module mounted on the main base unit of the control system CPU and makes access to the control system CPU.

2) If system switching occurs, the external device automatically continues access to the control system via the tracking cable. However, when the communication line with the connection destination is faulty or the standby system is powered off, for example, the connection destination must be changed on the external device side.

2) For access to the system that is not the connection destination (Example of connection to control system and access to standby system CPU)


External device
QnA-compatible 3E frame
・Request destination module I/O No. : 03D1H
・Network No. : 00H
・PLC No. : FFH
Access to standby system
Accessible if communication line is faulty
Control system (System A)
Standby system (System B)
Tracking cable

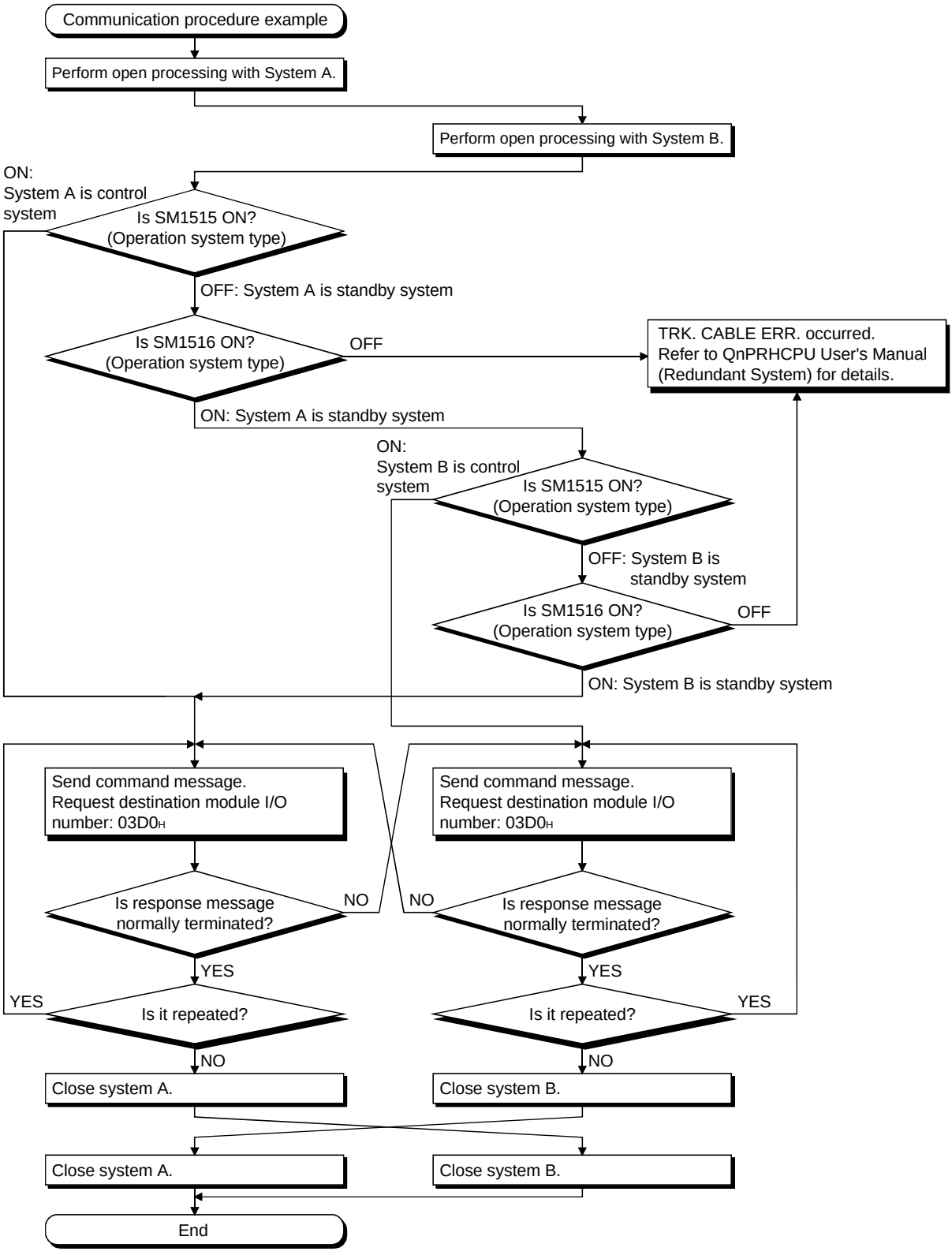
1) The external device connects to the Ethernet module mounted on the main base unit of the control system CPU and makes access to the standby system CPU via the tracking cable.
This enables access if the communication line between the external device and standby system is faulty.

5 - 102

(b) Communication procedure example for access to control system CPU in redundant system

The following shows a communication procedure example for access to the control system CPU in the redundant system.

Place the Ethernet module in an open waiting status and perform open/close processing from the external device.

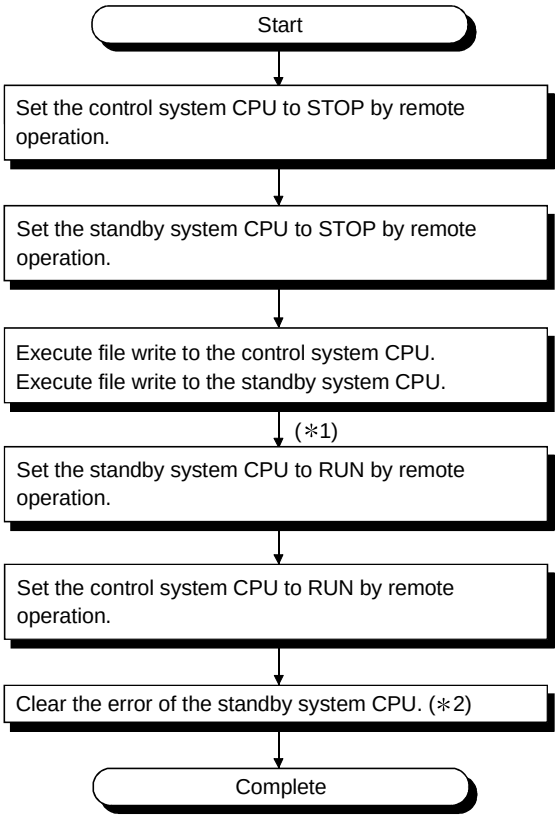


(c) Precautions for writing parameter file or program file

1) Be sure to write the same file to system A and system B.
An error will occur if different files are written or a file is written to only one system.

2) Write a parameter file or program file when the operating status of the CPU is "STOP".

3) Perform file write in the following procedure.


Start
Set the control system CPU to STOP by remote operation.
・A continue error occurs in the standby system CPU.
Set the standby system CPU to STOP by remote operation.
Execute file write to the control system CPU.
Execute file write to the standby system CPU.
(∗1)
Set the standby system CPU to RUN by remote operation.
Set the control system CPU to RUN by remote operation.
Clear the error of the standby system CPU. (∗2)
Complete

∗1 After writing a parameter file, simultaneously reset the both redundant CPUs.

∗2 When the operation status of the control system CPU is switched from STOP to RUN, check the error status of the standby system CPU, and if an error has occurred, set the error code (6010_H) to SD50, and then turn on SM50 to clear the error.

5 - 104

(4) "Communication via fixed buffer"

(a) Receive processing in standby system

When data is sent to the Ethernet module mounted on the standby system, the data received by the Ethernet module is discarded and data receive processing is not performed.
(The fixed buffer receive completion signal (corresponding bit of address: 5005H) does not turn on.)

(b) Receive processing in interrupt program

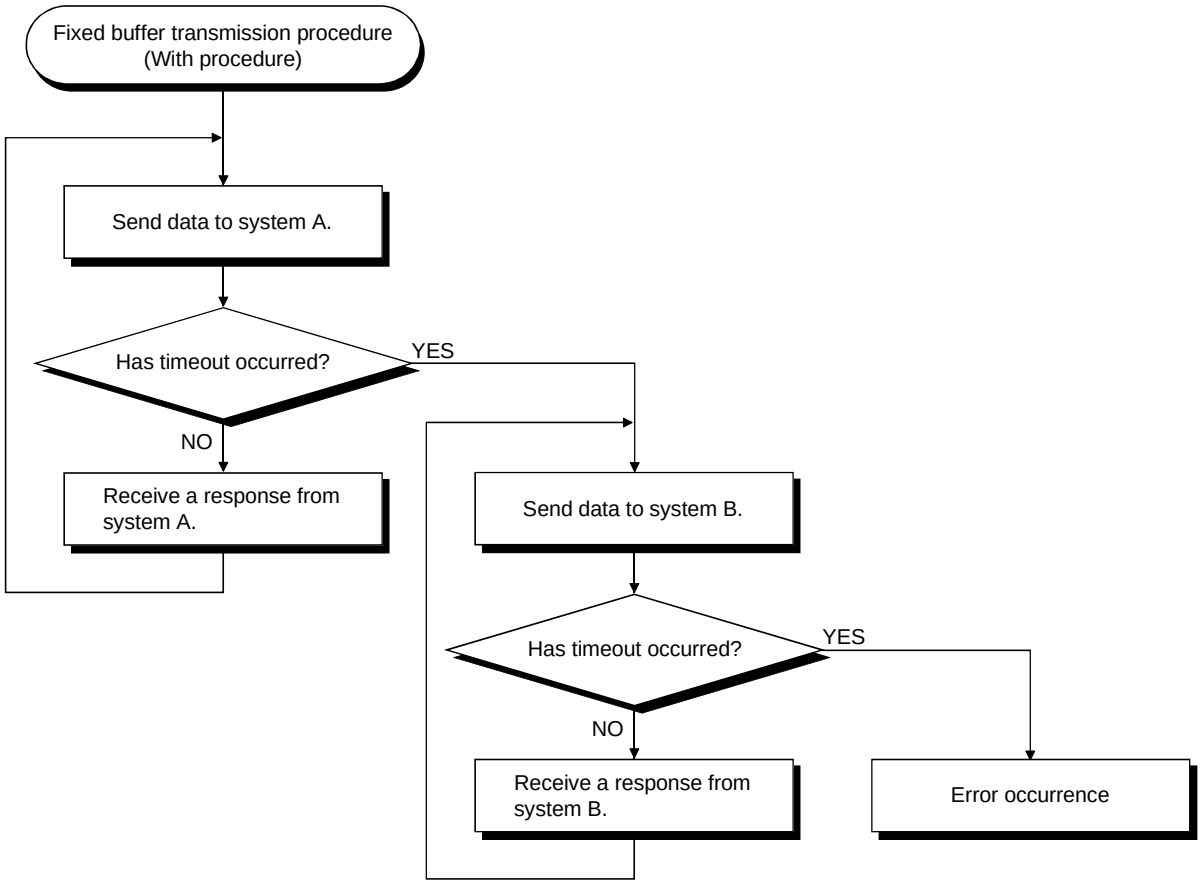
When the control system is switched to the standby system by of system switching before execution of an interrupt program, the interrupt factor is held.
When the system switching occurs again and the standby system is switched to the control system by the interrupt program is executed by the held interrupt factor.
(The interrupt factor is not transferred to the other system.)

(c) When data is sent from external device

1) When communication procedure is "with procedure"
If a response timeout to the Ethernet module occurs, change the connection destination to the other system and send data.

2) When communication procedure is "without procedure"
Send the same data to both the control system and standby system.

The following shows a transmission procedure example when data is transmitted to the redundant system via the fixed buffer (with procedure).



POINT

Note the following when executing re-transmission processing at system switching.

(1) When making communication while synchronizing transmission and reception
During communication, system switching may occur with transmission and reception not synchronized.
When system switching has occurred, resume communication after initializing synchronization to ensure safety.

(2) When using the dedicated instruction
Since whether the execution of the write instruction is completed or not is difficult to judge, the same instruction must be executed again.
In this case, note that the same instruction may be executed twice.

(5) "Communication via random access buffer"

The buffer memory of the Ethernet module is not tracked. Therefore, when writing data to the random access buffer, write the same data to the control system and standby system.

(6) "When using e-mail function"

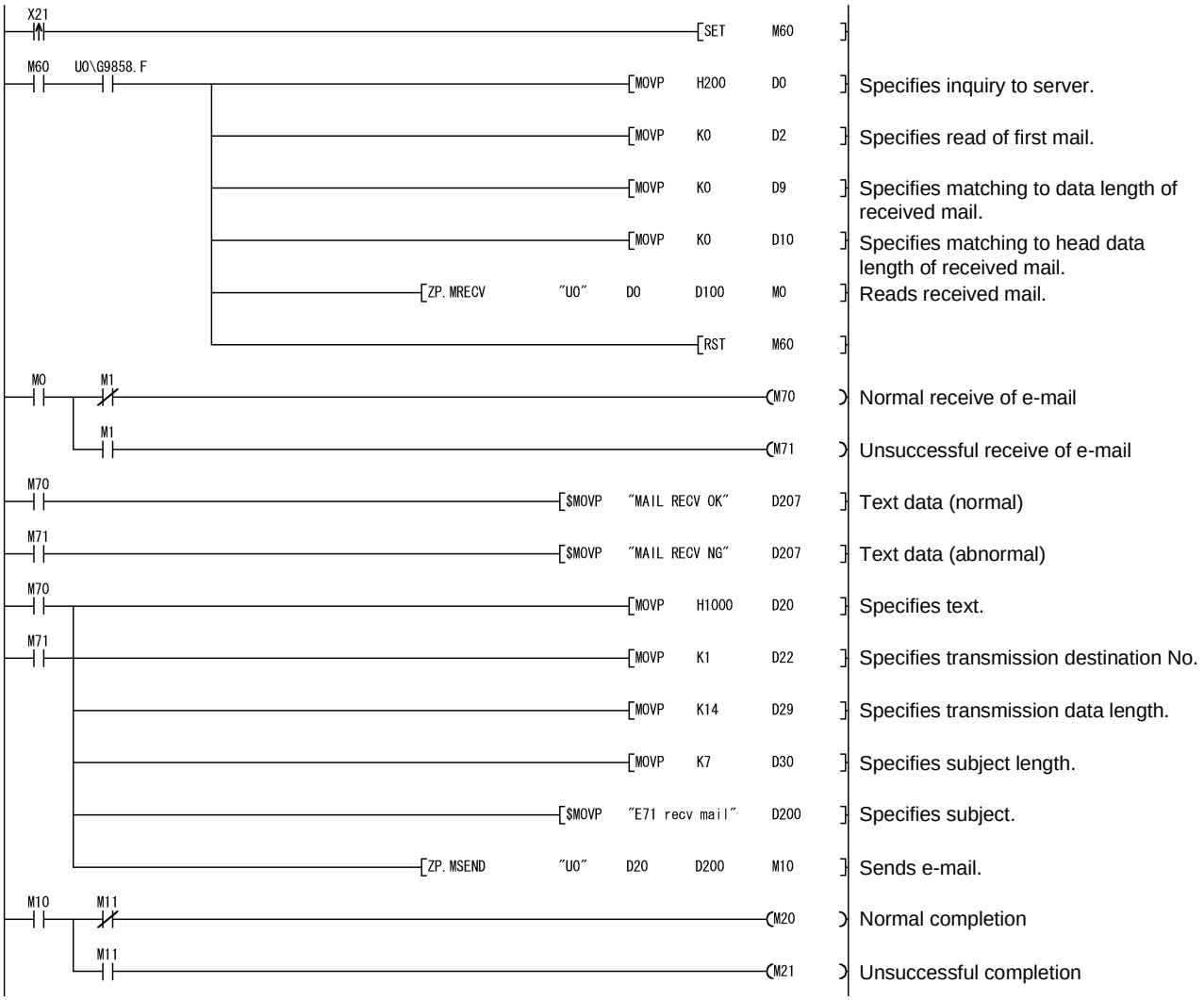
(a) Receive of e-mail

1) After receiving an e-mail by the Ethernet module, send a reply mail to the mail transmission source so that the mail transmission source confirms that it received the e-mail.
If receipt cannot be confirmed, send the e-mail again.

2) After the MRECV instruction is executed, the read e-mail is deleted from the mail server. Therefore, when system switching occurs during execution of the MRECV instruction, the mail may not be received by the new control system CPU after system switching if the MRECV instruction is re-executed. (The e-mail has been deleted from the mail server.)

(b) E-mail receive program example

In the following program, turning on X21 causes the Ethernet module mounted in the position of I/O signal X/Y00 to X/Y1F to receive an e-mail by execution of the MRECV instruction and then send a reply mail to the transmission source by execution of the MSEND instruction.
For the e-mail function, refer to Chapter 2 in the User's Manual (Application).



(c) When using reporting function

Since a reporting mail may be sent from both the control system and standby system in either of the following conditions, perform the processing that will discard the overlapping mail on the receive side personal computer. ∗1

- When the CPU operation status is set to the reporting condition
- When the device data set to the reporting condition is tracked

∗1 By setting the following SM devices as reporting conditions, the system of the redundant system can be identified by a reporting mail, and the devices can be used as conditions for identifying which system sent the overlapping mail.

- SM1511 (System A identification flag)
- SM1512 (System B identification flag)
- SM1515, SM1516 (Operation system type)

(7) "When communication is made via MELSECNET/H or MELSECNET/10"

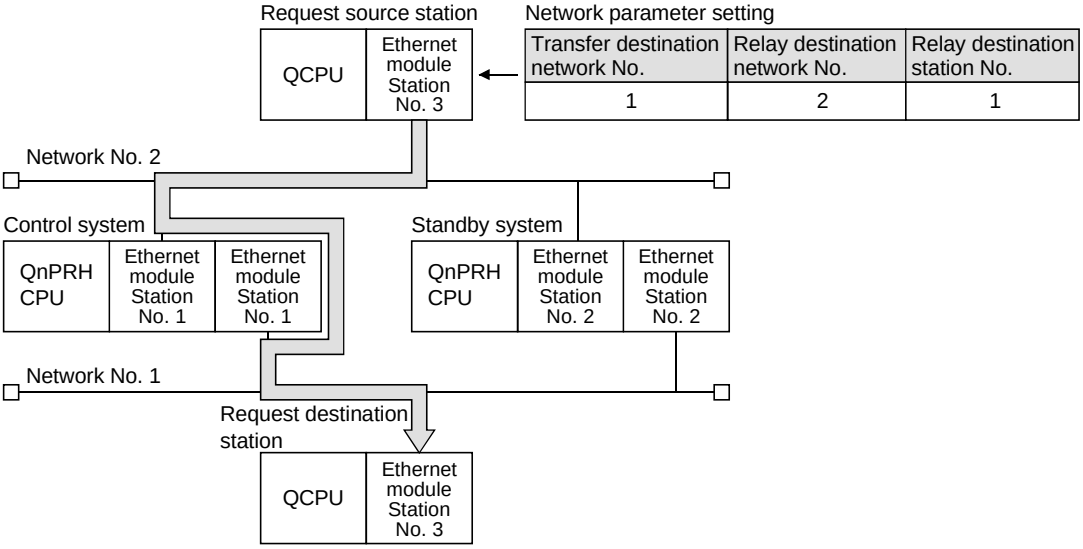
(a) When redundant system is on network

When making access via redundant system, note that the routing parameter setting must be changed by the RTWRITE instruction at the request source station or relay station at the time of system switching. (Refer to (b) below.)

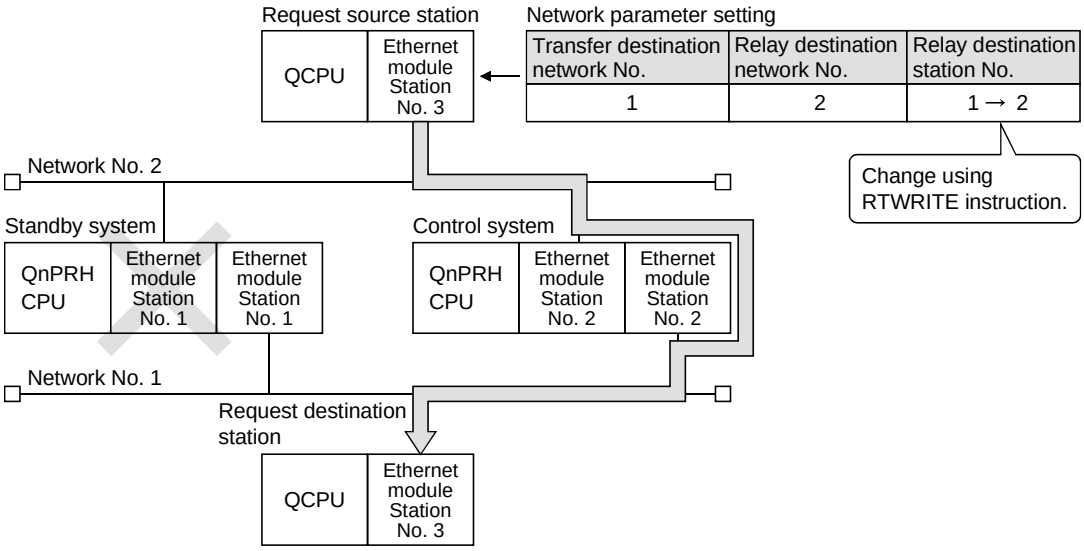
(b) When access is made via redundant system

The routing parameter setting must be set to the request source station or relay station to access other station via Ethernet using the MELSECNET/H, MELSECNET/10 relay communication function. (Refer to Chapter 3 in the User's Manual (Application).)

When making access via the redundant system, set the station, which becomes a control system, as a routing station.



Change the routing parameter setting with the RTWRITE instruction at the relay source station or relay station so that access is made via the station of the new control system after system switching when system switching occurs. (For the RTWRITE instruction, refer to the QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions).)

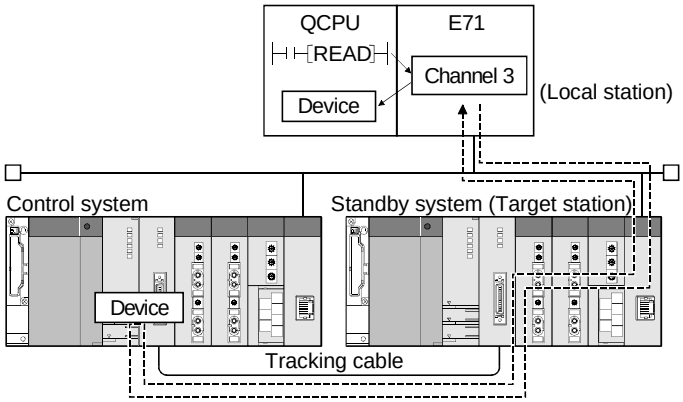


(8) "When QCPU accesses other station PLC using data link instruction"

(a) Redundant system-compatible data link instructions

1) By specifying the target station CPU type of the control data in any of the following data link instructions, access can be made to the control system/standby system or system A/system B. (Refer to Chapter 4 or Chapter 6 of the User's Manual (Application) for the data link instructions.)
   - READ/SREAD instruction
   - WRITE/SWRITE instruction
   - REQ instruction

2) Operation performed for access to control system/standby system or system A/system B (Example of executing the READ instruction)
   Since the received command is addressed to the control system CPU (target station CPU type: 3D0H) when the target station is in the standby system, data is relayed to the control system CPU via the tracking cable to read the data of the control system CPU.



(b) Processing at error completion

In case that the data link instruction is executed for the specified station (control system CPU/standby system CPU) in the redundant system and that target station causes system switching the data link instruction may result in error completion. (Error code: 4244H, 4248H)
If the data link instruction results in error completion due to the above error, execute the data link instruction again.

(c) SEND instruction

1) Execution of SEND instruction

When the target station is in a redundant system, the communication request source station must identify that the target station is the control system to execute the SEND instruction.

When the target station is the standby system, the target station storage channel cannot be used since the RECV instruction is not executed at the target station after data is sent by the SEND instruction. (Channel being used)

2) When performing broadcast

When a redundant system exists on the network where a broadcast is performed, the storage channel cannot be used since the RECV instruction is not executed for the standby system. (Channel being used)

(d) RECV instruction, interrupt program (RECVS instruction)

When the SEND instruction is executed for the redundant system, the processing of the RECV instruction or interrupt program (RECVS instruction) changes depending on the following condition.

1) When system switching occurs between execution of the SEND instruction for the control system and execution of the RECV instruction or interrupt program

When the control system is switched to the standby system by system switching before execution of the RECV instruction or interrupt program, it holds the RECV instruction execution request area (address: 205 (CD_H)) of the buffer memory or the interrupt factor (interrupt pointer) of the interrupt program.

When system switching occurs again and the standby system is switched to the control system, the RECV instruction or interrupt program is executed using the held RECV instruction execution request area of the buffer memory or the interrupt factor of the interrupt program.

2) When the SEND instruction is executed for the standby system

When the standby system receives data from the transmission station, it holds the RECV instruction execution request area (address: 205 (CD_H)) of the buffer memory and the interrupt factor (interrupt pointer) of the interrupt program.

Therefore, when the standby system is switched to the control system by system switching, the RECV instruction or interrupt program is executed using the held RECV instruction execution request area of the buffer memory and the interrupt factor of the interrupt program.

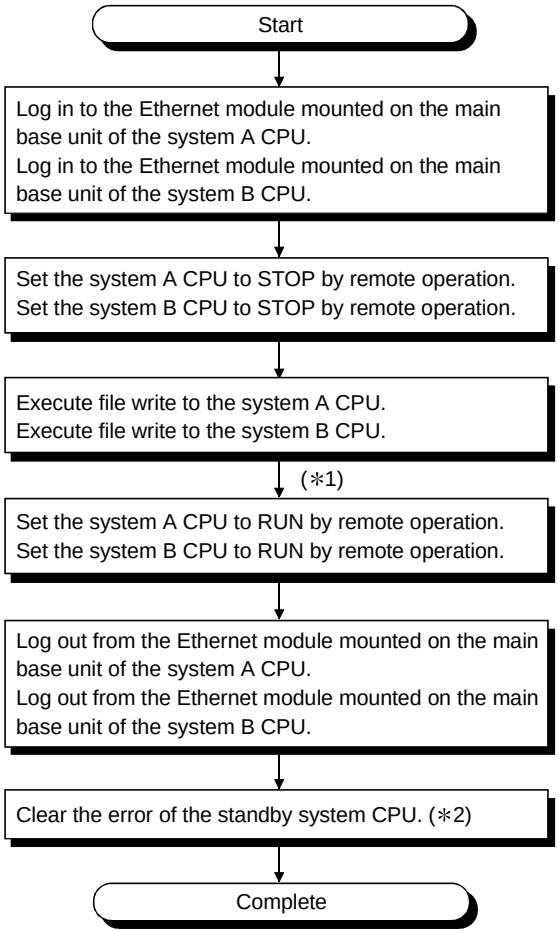
(9) "When using file transfer (FTP server) function"

(a) When writing parameter file or program file

1) Be sure to write the same file to system A and system B.
An error will occur if different files are written or a file is written to only one system.

2) Write a parameter file or program file when the operating status of the CPU is "STOP".

3) Perform file write in the following procedure.


Start
Log in to the Ethernet module mounted on the main base unit of the system A CPU.
Log in to the Ethernet module mounted on the main base unit of the system B CPU.
Set the system A CPU to STOP by remote operation.
Set the system B CPU to STOP by remote operation.
・A continue error occurs in the standby system CPU.
Execute file write to the system A CPU.
Execute file write to the system B CPU.
(∗1)
Set the system A CPU to RUN by remote operation.
Set the system B CPU to RUN by remote operation.
Log out from the Ethernet module mounted on the main base unit of the system A CPU.
Log out from the Ethernet module mounted on the main base unit of the system B CPU.
Clear the error of the standby system CPU. (∗2)
Complete

∗1 After writing a parameter file, simultaneously reset the both redundant CPUs.

∗2 When the operation status of the control system CPU is switched from STOP to RUN, check the error status of the standby system CPU, and if an error has occurred, set the error code (6010H) to SD50, and then turn on SM50 to clear the error.

(10) "When using dedicated instruction"

If system switching occurs during execution of the dedicated instruction, the dedicated instruction may not be completed.
In this case, execute the dedicated instruction again from the new control system CPU after system switching.

6 COMMUNICATION USING THE MC PROTOCOL

This chapter explains an overview of the MELSEC communication protocol (hereinafter referred to as the MC protocol).
See the following manual for details on the data communication function using the MC protocol:
MELSEC Communication Protocol Reference Manual (SH-080008)

6.1 Data Communication Function

The MC protocol is the abbreviated name of the MELSEC protocol that is a communication system for the Q series PLCs. Using this protocol, the external devices can read or write device data and programs from/to the PLC CPUs via the Q series Ethernet module or Q series serial communication module.
Any external devices on which application programs can be installed and which can send and receive data in accordance with the MELSEC PLC protocol can communicate with the PLC CPUs using the MC protocol.

6.1.1 Accessing the PLC CPUs using the MC protocol

This section explains the main functions for accessing the PLC CPUs using the MC protocol.
On the PLC side, the Ethernet module sends and receives data based on commands from the external devices.
Thus, the PLC CPU side does not require sequence programs for data communication.

(1) Data read/write

This function reads/writes data from/to the PLC CPU device memory of the local station or another station on the MELSECNET/H, MELSECNET/10 as well as the intelligent function module buffer memory.
By reading and writing data, the PLC CPU operation monitoring, data analysis and production management can be performed on the external device side.
Also, production instructions can be executed from the external device side.

(2) File read/ write

This function reads and writes files such as sequence programs and parameter files that are stored in the PLC CPU.
By reading and writing these files, the file management for the QCPU and the QnACPUs of other stations can be performed on the external station side.
Also, execution programs can be changed (replaced) from the external device side.

(3) Remote control of the PLC CPU

This function executes remote RUN/STOP/PAUSE/latch clear/reset operations. Remote operations of the PLC CPU can be performed from the external device side using the PLC CPU remote control function.

	Function		
Communication using the MC protocol	Communication using QnA compatible 3E frames		Communication using ASCII code
	Communication using A compatible 1E frames		Communication using binary code
	Device memory read/write	Batch read/write in bit/word units	
		Device memory monitoring	
		Multiple block batch read/write	
		Read/write by extension designation	
		Other station access via a network system	
	Read/write of the Ethernet module buffer memory		
	Read/write of the intelligent function module buffer memory		
	Read/write of the sequence program files		
	Status control of the PLC CPU (remote RUN/STOP, etc.)		

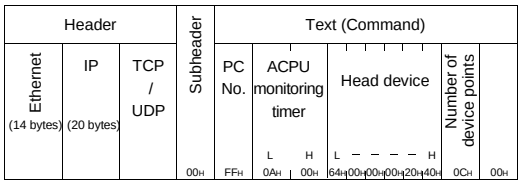
6.1.2 Message format and control procedure for data communication

The data communication functions using the MC protocol correspond to the functions for reading/writing data in the PLC CPU supported by the A/QnA series Ethernet interface modules.
Therefore, the message format and control procedure are the same as when accessing the PLC using A/QnA series Ethernet interface modules.

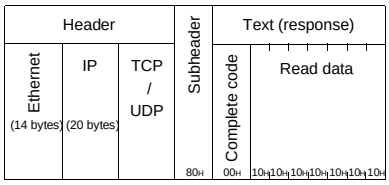
- Communication using QnA compatible 3E frames
  This message format is used for the QnA series Ethernet interface modules
- Communication using A compatible 1E frame
  This message format is used for the A series Ethernet interface modules

The external device side can access the Q series PLC with a program used for accessing the PLC via A/QnA series Ethernet interface module.

(Example)



(Command message for the A compatible 1E frame)



(Response message for the A compatible 1E frame)

POINT

The following manual is available for performing data communication using the MC protocol.
Q Corresponding MELSEC Communication Protocol Reference Manual (sold separately)

6.1.3 PLC CPU setting for performing data communication

Data communication via MC protocol is enabled by making the following settings using GX Developer and writing the parameters to the PLC CPU.

1) Setting the number of Ethernet cards (see Section 4.6)
2) Initial settings (see Section 5.2)
3) Open settings (see Section 5.5)

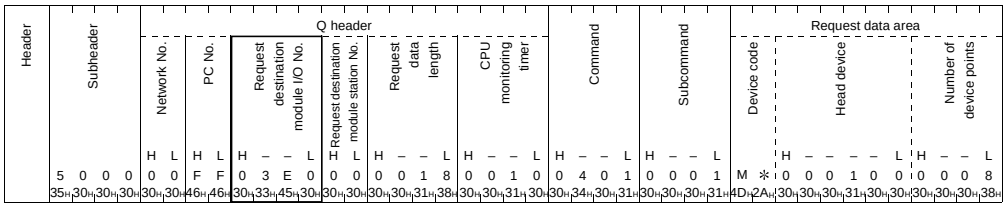
POINT

(1) By using the automatic open UDP ports of the Ethernet module, communication using the MC protocol can be performed regardless of the RUN/STOP status of the PLC CPU.
When the automatic open UDP ports are not used, connect the user open port first and perform data communication referring to Chapter 4, "Settings and Procedures Prior to Starting the Operation" and Chapter 5, "Communication Procedure" of this manual.

(2) When writing from an external device to the PLC CPU, the enable/disable write at CPU RUN time setting can be set using the GX Developer "Ethernet operational settings" parameters.
* When writing data to the remote I/O station of the MELSECNET/H, set the enable/disable write at CPU RUN time parameter to "Enable".

6.1.4 Support of multiple CPU system or redundant system

When the external device accesses the QPU in a multiple CPU system or redundant system, specifying the target QCPU with the "request destination module I/O number" in the QnA-compatible 3E frame for MC protocol enables access to the control CPU or non-control CPU of the multiple CPU system or to the control system CPU/standby system CPU or system A CPU/system B CPU of the redundant system.
Refer to the Reference Manual for details.
Refer to Chapter 2 of this manual for the system configuration.
(Example) When multiple system CPU No. 1 is specified



(Command message for QnA compatible 3E frame)

REMARK

- When using the Ethernet module in a multiple CPU system, set the QCPU controlling the Ethernet module (hereinafter referred to as the control CPU) through GX Developer.
- It is possible to mount an Ethernet module of function version A on a multiple CPU system, in which case the access is limited to the control PLC (PLC No.1).

6.1.5 Support for the QCPU remote password function

When the following parameters have been set for the Ethernet module mounted on the QCPU, the Ethernet module performs the remote password check when the external device accesses the PLC:
(QCPU parameter settings)

- When a remote password is set in the QCPU
- When the connection that is communicating data with the external device is set as a target for the remote password check

Section 5.9 of this manual explains the remote password subject to the password check, data communication procedure and the unlock/lock processing for the remote password.
Refer to Section 5.9 first if the connection that is communicating data with the external device is set as a target for the remote password check.

6.2 Utilizing the MX Component

If the external device is a PC running one of the basic operation systems below, it is possible to create a communication program for the external device without considering the detailed MC protocol (transmission/reception procedures) using one of the following separately sold communication support tools.
Refer to Appendix 9 for the overview of MX Component.

(Supported basic operation systems)
- Microsoft® Windows® 95 Operating System
- Microsoft® Windows® 98 Operating System
- Microsoft® Windows NT® Workstation Operating System Version 4.0
- Microsoft® Windows® Millennium Edition Operating System
- Microsoft® Windows® 2000 Professional Operating System
- Microsoft® Windows® XP Professional Operating System
- Microsoft® Windows® XP Home Edition Operating System

✻ Depending on the version of MX Component used, different operating systems are supported.
See the manual of MX Component for details.

(Communication support tools)
- MX Component (SW0D5C-ACT-E or later.)

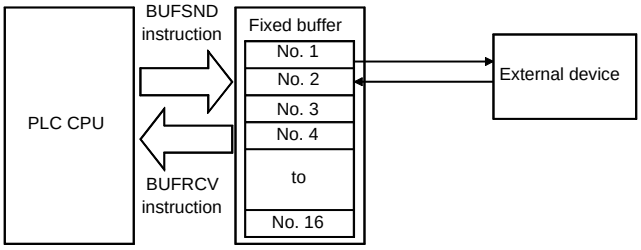
7 FIXED BUFFER COMMUNICATION (WITH THE PROCEDURE EXIST CONTROL METHOD)

This chapter explains how the PLC CPU and external device communicate in a 1:1 mode using the fixed buffers (with the procedure exist control method) of the Ethernet module.

7.1 Control Method

The following explains how communication is performed using the fixed buffers and the procedure exist control method.
In the communication processing using the fixed buffers, data transmission from the PLC CPU and the external device is executed through handshaking.

(1) The data flow in the communication processing is as follows.

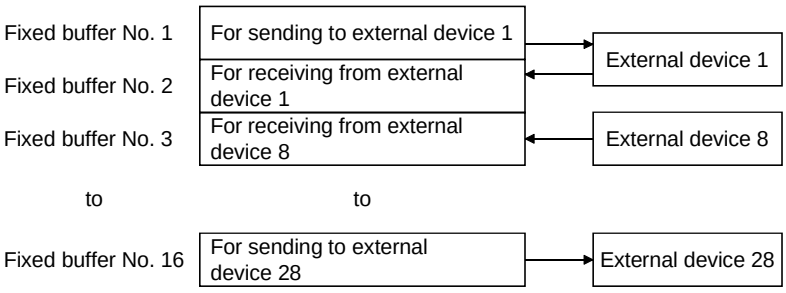


(2) Data can be communicated with the following external devices.
- Device on the Ethernet to which the Ethernet module is connected.
- Devices connected with the router relay function (see Section 5.3)

As shown in the diagram below, when using each fixed buffer (No. 1 to 16), the destination devices and usage conditions (for sending/receiving, procedure exist/no procedure, etc.) should be set when the connection via the Ethernet module is opened to fix the external device for each buffer.

(a) At TCP/IP communication
It is allowed to change external devices only when the open completion signal of the applicable connection is off.

(b) At UDP/IP communication
External devices can be changed regardless of the status of the applicable connection.
("Destination IP address" and "Destination Port No." in the communication address setting area can be changed. However, the "Local station Port No." cannot be changed.)
When changing external devices, do not use the "Pairing open" and "Existence confirmation" functions.



7 - 1

POINT

In communication where the procedure exist control method is selected, the data can be communicated by the following methods after the open processing is completed.

- Fixed buffer communication with the procedure exist control method (sending or receiving)
- Communication using the random access buffers
- Communication using the MC protocol

(3) At data sending/receiving, the Ethernet module performs the following processing.

(a) At data sending

When the PLC CPU executes the dedicated BUFSND instruction (*1) in a sequence program, the Ethernet module sends data of the applicable fixed buffer (No. n) to the external device that is specified in the communication address setting area (addresses: 28H to 5FH and 5038H to 507FH) corresponding to fixed buffer No. n (*2).

(b) At data receiving

The Ethernet module processes the received data if the data is received from an external device set in the communication setting area that corresponds to fixed buffer No. n. (*2)

If data is received from an external device not set in the connection information area of the buffer memory, the Ethernet module does not request to read the receive data to the PLC CPU side.

*1 For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

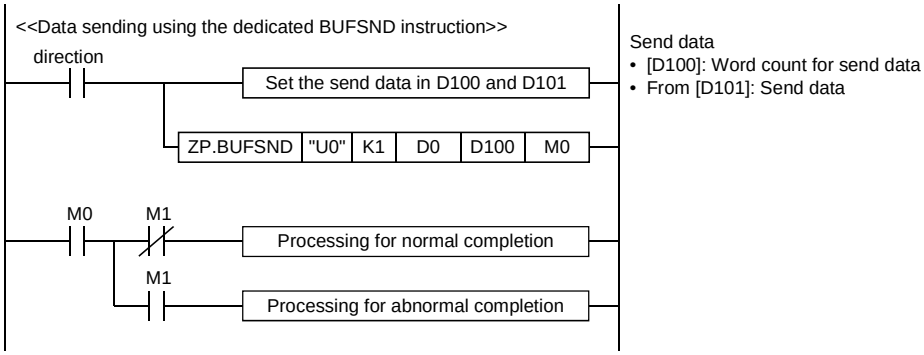
*2 In case of TCP/IP Unpassive open, data is communicated with an external device stored in the connection information area corresponding to fixed buffer No. n.

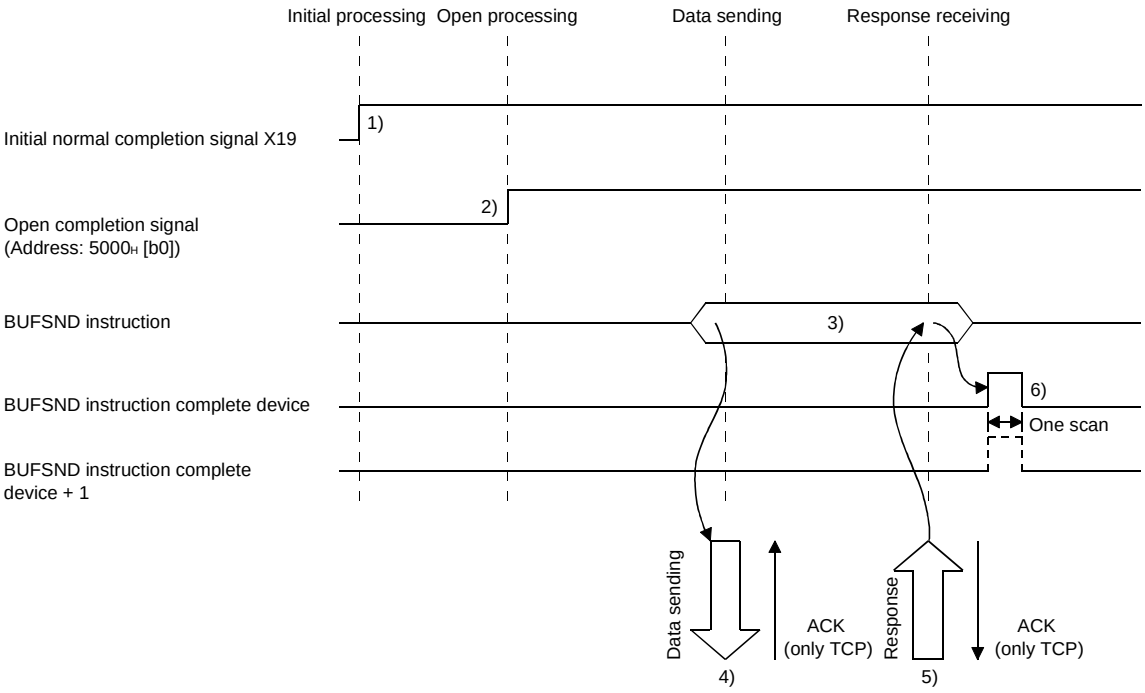
POINT

When the Ethernet module stores the receive data in the corresponding fixed buffer in the receive processing for the simultaneous broadcasting, it updates the destination IP address and destination port No. in the connection information area (addresses: 78H to C7H and 5820H to 586FH) that corresponds to fixed buffer No. n.

7.2 Sending Control Method

This section explains the control method when data is sent from the Ethernet module to an external device using the fixed buffer No. 1 and the area corresponding to connection No. 1 as an example.





7 - 3

1) Confirm the normal completion of the initial processing.
2) Confirm the normal completion of the open processing of connection No. 1
3) Execute the dedicated BUFSND instruction.
   The Ethernet module processes the following and sends the data.
   • Writes the send data length and send data to the fixed buffer (No. 1) area.
     Send data length : The head address area of the target fixed address (∗1)
     Send data : Area beginning from the head address of the target fixed buffer + 1
   ∗1 The send data length denotes a word count.
4) The size of send data in the fixed buffer (No. 1) area designated by the send data length is sent to the designated external device (set in the open processing).
5) Upon receiving the data from the Ethernet module, the external device returns a "Response" to the Ethernet module.
6) Upon receiving the "Response" from the external device, the Ethernet module ends the data transmission.
   If the "Response" is not returned within the response monitoring timer values (see Section 5.2), a data send error occurs.

   At normal completion
   • BUFSND instruction complete device : ON
   • BUFSND instruction complete device + 1 : OFF
   • BUFSND instruction complete status area (∗2) : 0000H
   • Response end code (∗3) : 00H

   At abnormal completion
   • BUFSND instruction complete device : ON
   • BUFSND instruction complete device + 1 : ON
   • BUFSND instruction complete status area (∗2) : Value other than 0000H
   • Response end code (∗3) : Value other than 00H

   After the data transmission is abnormally completed, execute the dedicated BUFSND instruction again to repeat the transmission processing.

∗2 The status at completion is stored in the complete status area of the BUFSND instruction. For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".
∗3 The response end code is stored in the communication storage status area of the buffer memory.
   For details on the response end code, see Section 7.4.2, "Application Data, (5) End codes".

POINT

The destination setting (see Section 5.5) for a connection whose parameters are set with GX Developer becomes valid when the open completion signal (address: 5000H … corresponding bit) of the Ethernet module switches from off to on.

7.3 Receiving Control Method

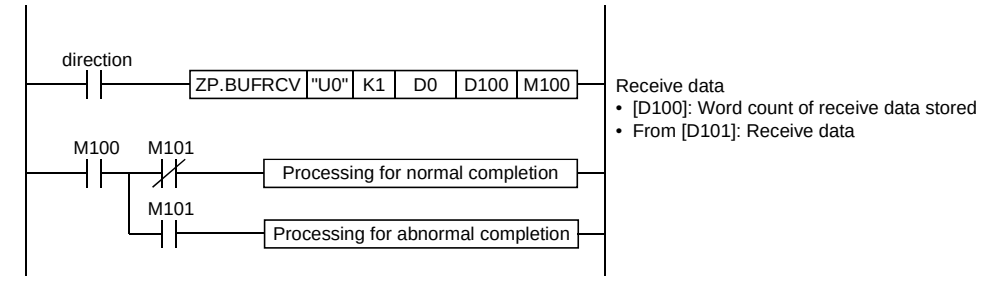
This section explains the control method when the Ethernet module receives data from an external device.
Fixed buffer communication employs the following receive processing methods:

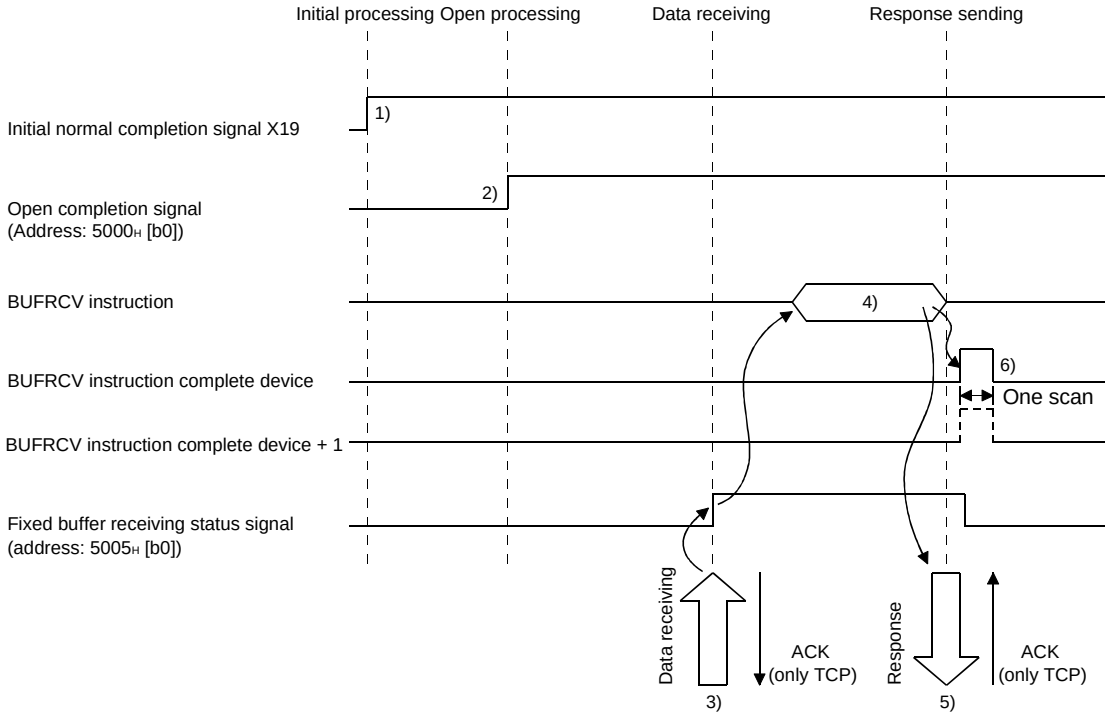
- Receive processing with the main program (dedicated instruction: BUFRCV)
  : See Section 7.3.1
- Receive processing with an interrupt program (dedicated instruction: BUFRCVS)
  : See Section 7.3.2

7.3.1 Receive processing with the main program (dedicated instruction: BUFRCV)

This section explains about the receive processing to be performed with the main program, using an example in which the fixed buffer No. 1 and the area corresponding to connection No. 1.

<<Data receiving using the dedicated BUFRCD instruction (main program)>>





1) Confirm the normal completion of the initial processing.

2) Confirm the normal completion of the open processing of connection No. 1.

3) Upon receiving data from the designated external device (set in the open processing), the Ethernet module processes the following.
   - Stores the receive data to the fixed buffer (No. 1) area.
     Receive data length : The head address area of the target fixed address
     Receive data : Area beginning from the head address of the target fixed buffer + 1
   - Fixed buffer receive status signal (address: 5005H … b0) : ON

4) Execute the dedicated BUFRCV instruction to read the receive data length and receive data from the fixed buffer (No. 1).

5) When the receive data length and reception data are read, the following processing is performed.

   At normal completion
   - Return "Response" to communication destination.
   - BUFRCV instruction complete device : ON
   - BUFRCV instruction complete device + 1 : OFF
   - BUFRCV instruction complete status area (*1) : 0000H

   At abnormal completion
   - Return "Response" to communication destination.
   - BUFRCV instruction complete device : ON
   - BUFRCV instruction complete device + 1 : ON
   - BUFRCV instruction complete status area (*1) : Value other than 0000H

6) End the receive processing.

*1 The status at completion is stored in the complete status area of the dedicated instructions. For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

POINT

(1) The destination setting (see Section 5.5) for a connection whose parameters are set with GX Developer becomes valid when the open completion signal (address: 5000H … corresponding bit) of the Ethernet module switches from off to on.

(2) Execute the BUFRCV instruction when the corresponding connection's bit in the fixed buffer receive status signal storage area (address: 5005H) of the buffer memory switches from off to on.

(3) At abnormal data receiving, the fixed buffer receive completion signal (address: 5005H … b0) does not turn on.
In addition, data is not stored in the fixed buffer (No. 1) area.

7.3.2 Receive processing with an interrupt program (dedicated instruction: BUFRCVS)

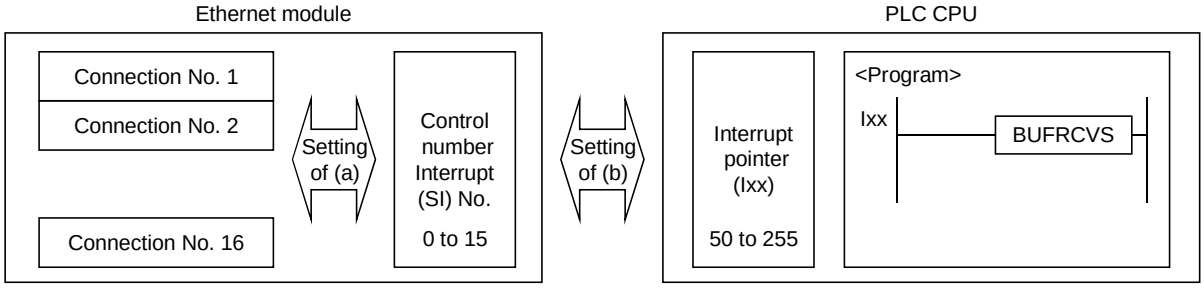
This section explains about the receive processing when an interrupt program is used. When an interrupt program is set to handle the receive processing, the interrupt program starts up when data is received from an external device and reading the receive data destined for the PLC CPU is enabled.

In order to start up the interrupt program, set the parameters using GX Developer.

(1) Setting screen

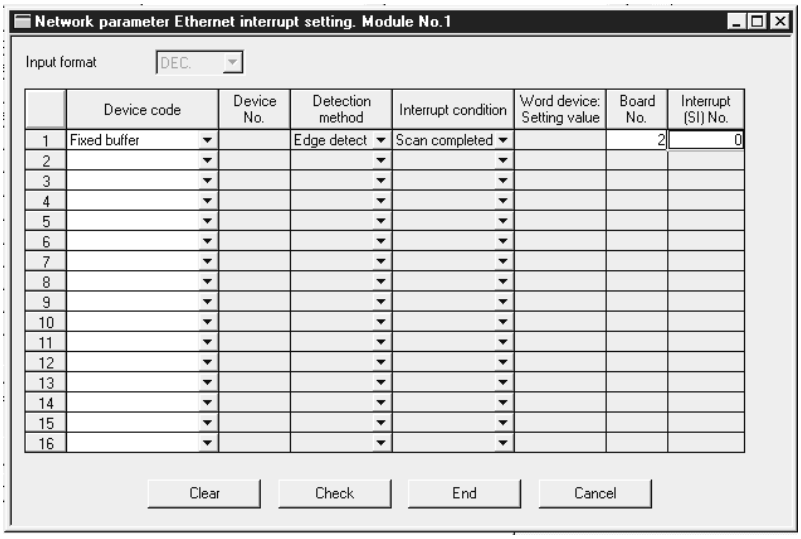
Set the following parameters using GX Developer to start the interrupt program.

- "Network parameters" — "Interrupt settings"
  Set the control number (SI) on the Ethernet module side when an interrupt request is issued from the Ethernet module to the PLC CPU.
- "PLC parameters" — "Interrupt pointer settings"
  Associate the control number (SI) set in "Network parameters" — "Interrupt settings" with the interrupt pointer (Ixx) used in the interrupt program.



(a) "Network parameters" - "Interrupt settings"

The interrupt settings using GX Developer are explained below.
Start the "Ethernet interrupt settings" screen by selecting "Setting the number of MNET/10H Ethernet cards" - "Interrupt settings".



Item name	Description of setting	Setting range/option
Input format	Select the input format of each data.	• Decimal • Hexadecimal
Device code	Select device code.	• Fixed buffer • RECV instruction
Device No.	—	—
Detection method	—	(Edge detection)
Interrupt condition	—	(Scan complete)
Word device setting value	—	—
Board No.	Set channel number/connection number used in the interrupt program.	• RECVS instruction 1 to 8 • BUFRCVS instruction 1 to 16
Interrupt (SI) No.	Set the interrupt No.	0 to 15

1) Input format
   - Select the input format (decimal/hexadecimal) of each setting item.

2) Device code
   - Select "Fixed buffer".
   - The interrupt program starts up when the receive data is stored in the fixed buffer of the connection that opened the port set in "3) Board No." below.

3) Board No.
   - Set the connection No. (1 to 16) of the fixed buffer that will initiate the startup of the interrupt program.

4) Interrupt (SI) No.
   - Set the interrupt control No. (0 to 15) on the Ethernet module side when the interrupt request is issued from the Ethernet module to the PLC CPU.
   - Set a unique interrupt (SI) No. that does not overlap with the ones for interrupts of other fixed buffer communication and the RECV instruction.

   ✻ The interrupt (SI) No. (0 to 15) can be arbitrarily assigned by the user to receive data in communication using up to 16 fixed buffers as well as to receive data using the RECV instruction by the interrupt program. The user must manage the interrupt (SI) No. assigned to each data reception method.

   (Example)

   For receiving by the fixed buffer communication, assign the same interrupt (SI) No. as the target data communication connection.
   For receiving using the RECV instruction, assign an interrupt (SI) No. that is not assigned to the fixed buffer communication.

REMARK

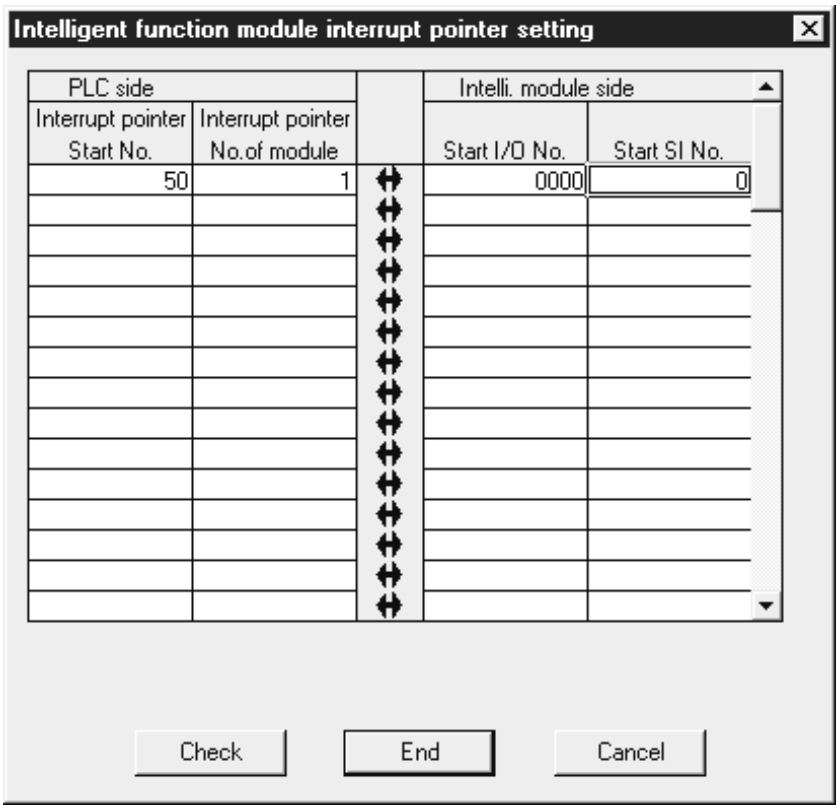
Items other than the ones mentioned above do not need to be set by the user on the "Ethernet interrupt settings" screen.
The setting values shown in the table above are automatically displayed for items related to the detection method and event conditions.

POINT

In order to start up the interrupt program, the "Network parameter Ethernet interrupt setting" and "PLC parameters" – "Intelligent function module interrupt pointer setting" are required.

(b) "PLC parameters" - "Interrupt pointer settings"

The interrupt pointer settings using GX Developer are explained below.
Start the "Intelligent function module interrupt pointer setting" screen by selecting "PLC parameters" - "PLC system settings" - "Interrupt pointer settings."



Item name		Description of setting	Setting range/option
PLC side	Interrupt pointer Start No.	Set the interrupt pointer Start No. on the PLC CPU side.	50 to 255
	Interrupt pointer No. of module	Set the interrupt pointer count on the PLC CPU side.	1 to 16
Intelli. module side	Start I/O No.	Set the module's starting I/O No.	0000 to 0FE0H
	Start SI No.	Set the interrupt pointer (SI) starting No. on the PLC CPU.	0 to 15

1) PLC side - Interrupt pointer Start No.
   - Set the start No. (50 to 255) of the interrupt pointers (Ixx) used in the sequence program.
2) PLC side - Interrupt pointer No. of module
   - Set the number (1 to 16) of interrupt pointers (Ixx) used in the sequence program.
3) Intelli. module side - Start I/O No.
   - Set the start I/O No. of the Ethernet module.
4) Intelli. module side - Start SI No.
   - Set the smallest number (0 to 15) among the maximum 16 interrupt (SI) No. (the control numbers on the Ethernet module side) set in "(a) Interrupt settings".

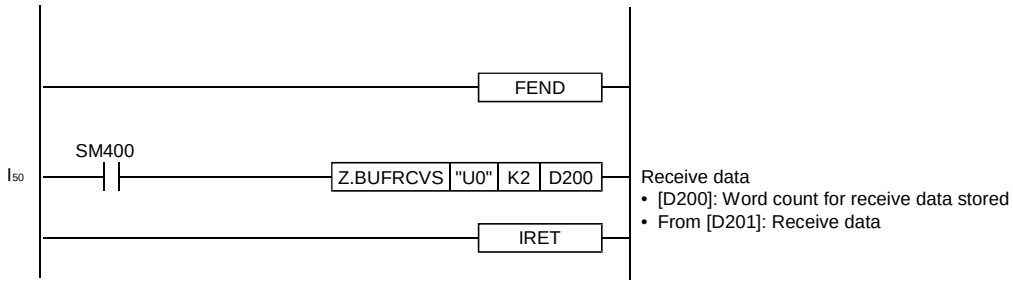
POINT
In order to start up the interrupt program, the "Network parameters Ethernet interrupt setting" and "PLC parameters" – "Intelligent function module interrupt pointer setting" are required.

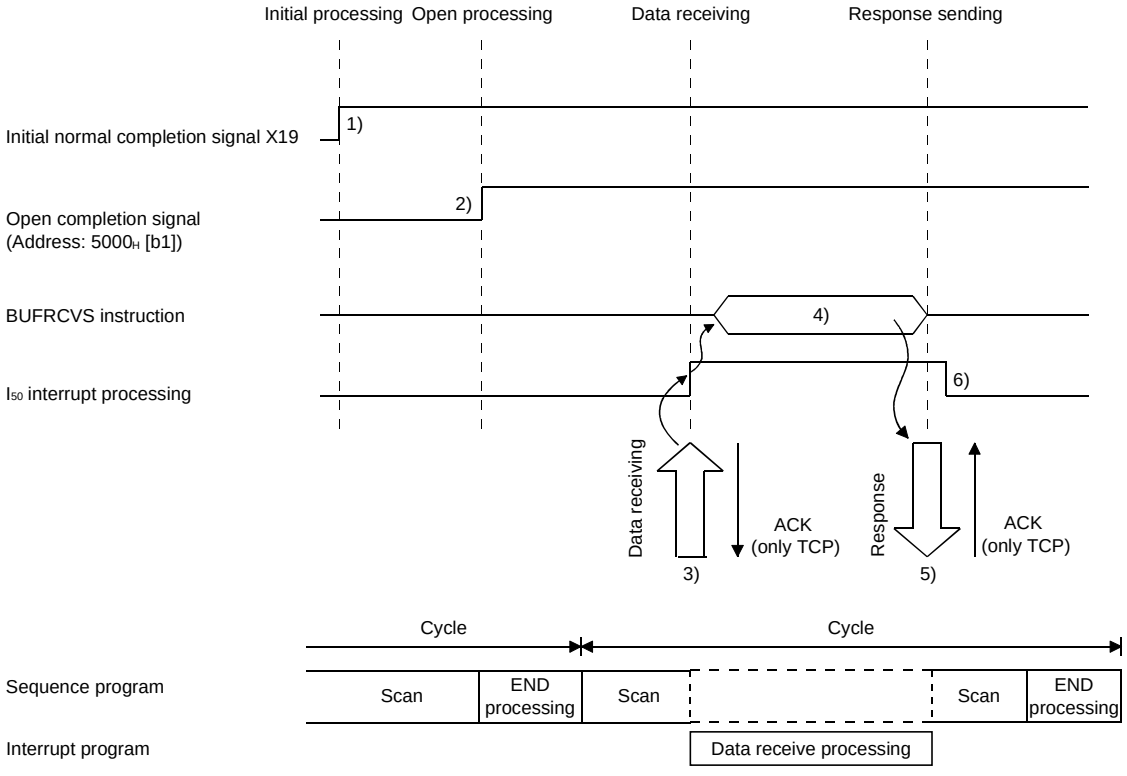
(2) Control method

The control method when the interrupt program starts up is explained using fixed buffer No. 2 and the area corresponding to connection No. 2 as an example.

* The interrupt settings with GX Developer for reading receive data using the interrupt program example below is illustrated in the screen display shown in (1).

<<Data receiving using the dedicated BUFRCVS instruction (interrupt program)>>





1) Confirm the normal completion of the initial processing.

2) Confirm the normal completion of the open processing of connection No. 2.

3) Upon receiving data from the designated external device (set in the open processing), the Ethernet module processes the following.
   - Stores the receive data to the fixed buffer (No. 2) area.
     Receive data length : The head address area of the target fixed address
     Receive data : Area beginning from the head address of the target fixed buffer + 1
   - Fixed buffer receive status signal (address: 5005H … b1) : ON
   - Request the PLC CPU to start up the interrupt program.

4) The interrupt program starts up.
   Execute the dedicated BUFRCVS instruction to read the receive data length and receive data from the fixed buffer (No. 2).

5) When the receive data length and reception data are read, the following processing is performed.

   At normal completion
   - Return a "Response" to the destination.
   - PLC CPU error flag (SM0) (*1) : OFF

   At abnormal completion
   - PLC CPU error flag (SM0) (*1) : ON
   - PLC CPU error code (SD0) (*1) : error code

6) The execution of the interrupt program is finished and the execution of the main program starts again.

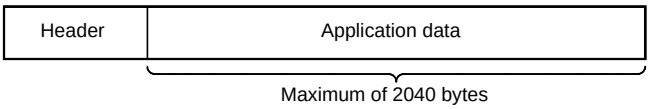
   *1 For information of the PLC CPU error flag (SM0) and error code (SD0), see the manual for the PLC CPU.

REMARK

To start up an interrupt program, create an interrupt enable/disable program in the main program.
The instructions to be used for this are EI and DI.

7.4 Data Format

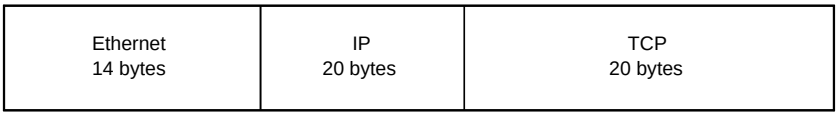
When communicating data between the Ethernet module and an external device, the data format explained below is used.
The communication data consists of a "header" and "application data" as follows:



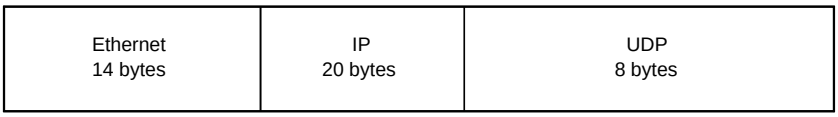
7.4.1 Header

The header for TCP/IP or UDP/IP is used. In case of the Ethernet module, the Ethernet module adds and deletes the header. Thus, the user does not need to set it.
(Details of the size of the header section)

1) In case of TCP/IP



2) In case of UDP/IP

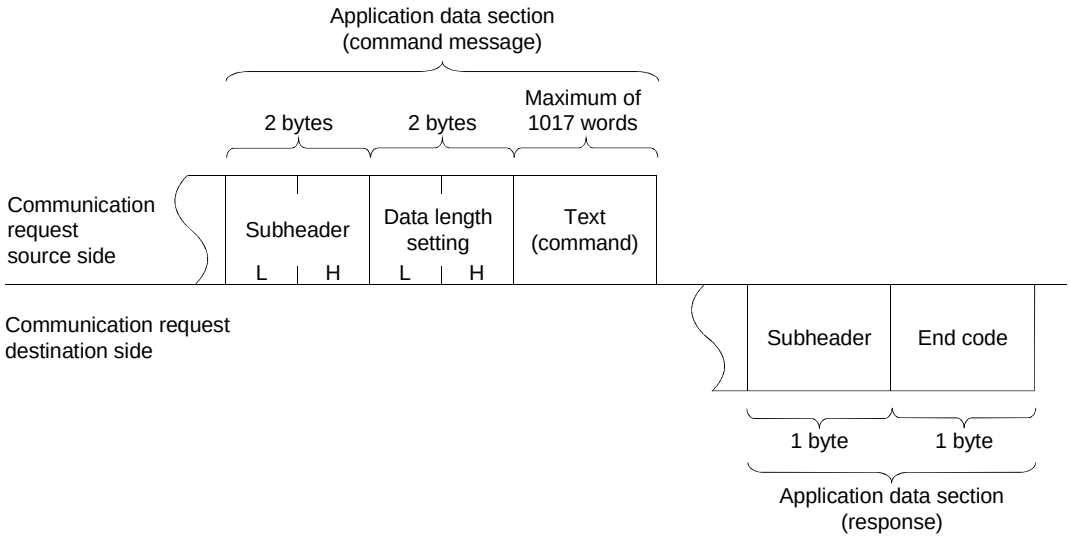


7.4.2 Application data

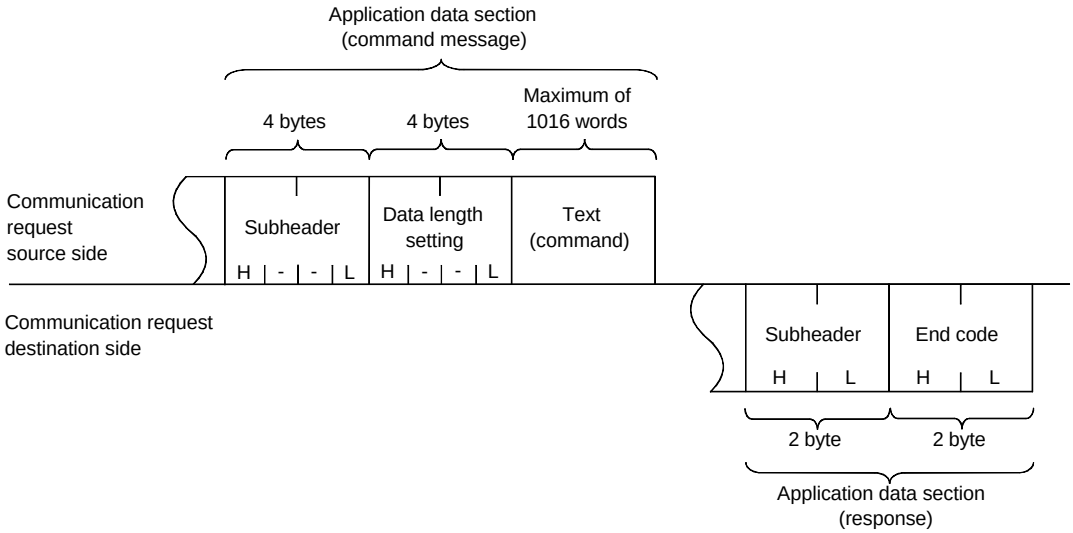
As shown below, the data code in the application data can be expressed in either binary or ASCII code. Switching between binary code and ASCII code is performed with GX Developer as follows.
[GX Developer] – [Network parameters] ― [Operational settings] ― [Communication data code]
For more details, see Section 4.7, "Operational Settings".

(1) Format

(a) Communication using binary code



(b) Communication using ASCII code

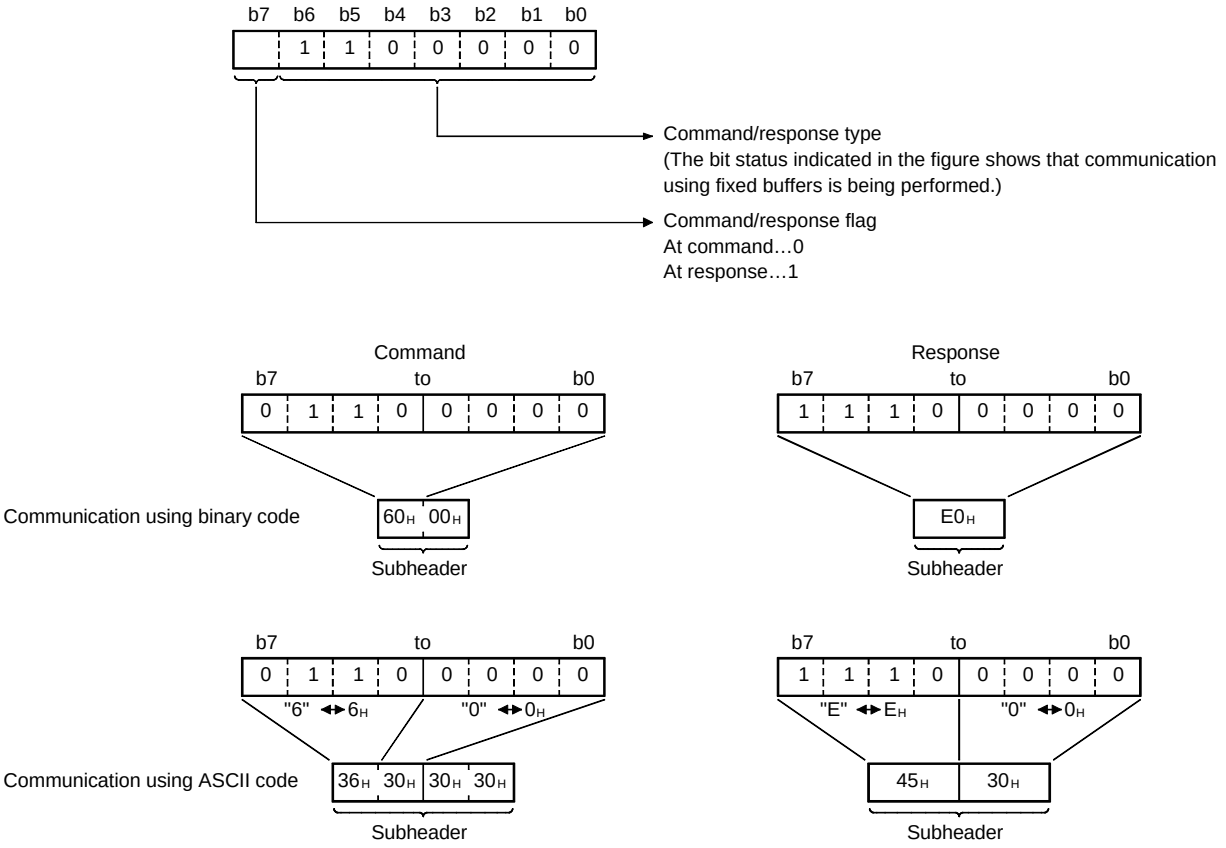


7 - 13

(2) Subheader

The format of the subheader is as shown below.

The user does not need to set the subheader when using the Ethernet module since the Ethernet module adds and deletes it.



7 - 14

(3) Data length setting

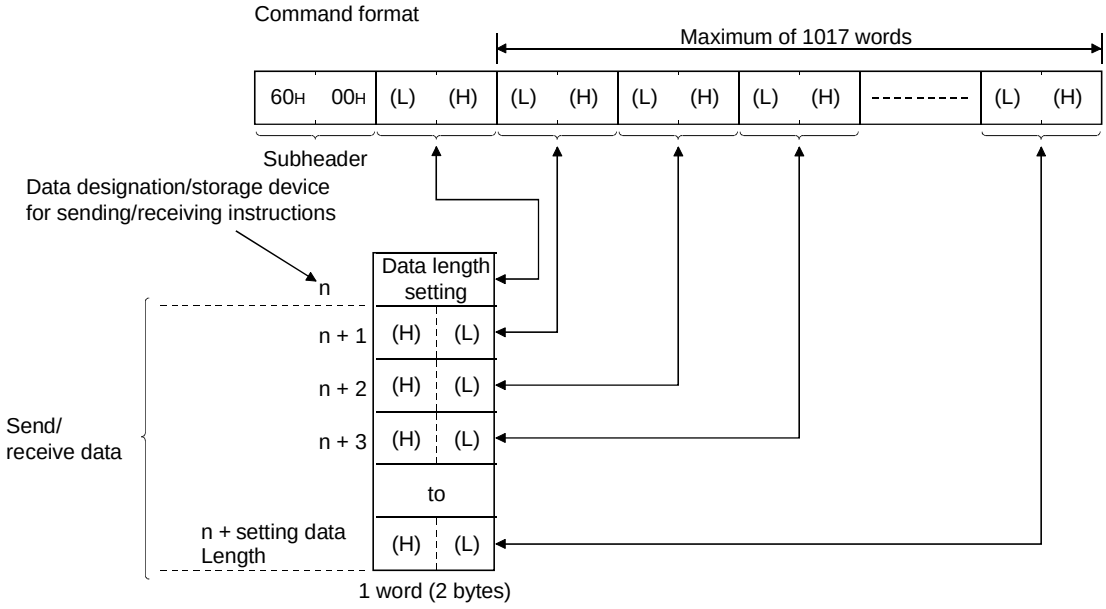
This value indicates the data size of the text (command) section.

POINT
The data length can be designated in the following range: • Communication using binary code: Maximum of 1017 words • Communication using ASCII code: Maximum of 508 words (*1) *1 Since data is sent/received as ASCII data, the communication data size is approximately half of the data size when using binary code.

(4) Text (command)

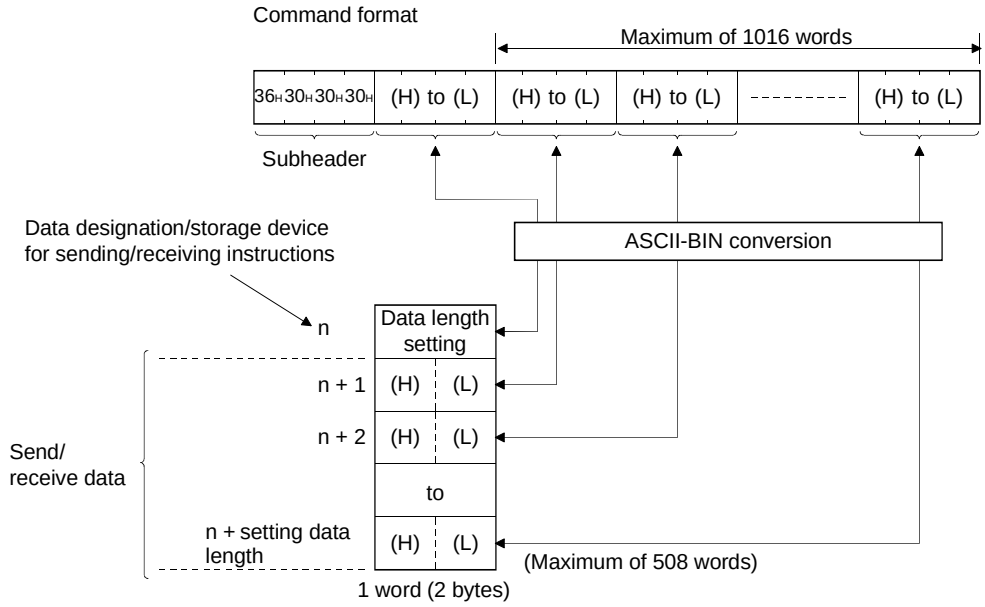
The format of the command/response when communicating using fixed buffers is configured as follows.

(a) Communication using binary code

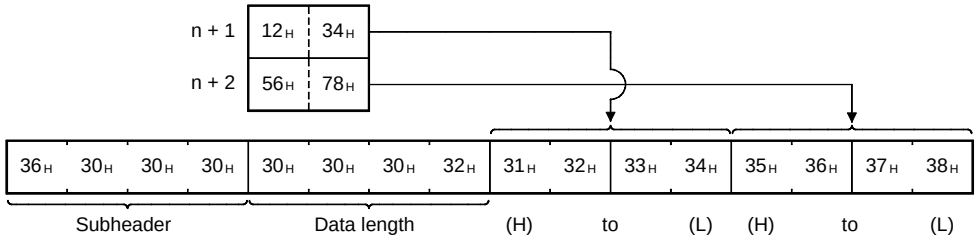


7 - 15

(b) Communication using ASCII code



(Example)



(5) End codes

For more details on the end codes added to a response when communicating using the fixed buffers, see Section 11.3.1.

End codes are stored in the complete status area (in the control data) of the BUFSND and BUFRCV instructions, as well as the communication status storage area of the buffer memory.

7.5 Programming

This section explains the programming method when the Ethernet module and an external device communicate using the fixed buffers and the procedure exist control method.

7.5.1 Precautions when creating programs

(1) In order to communicate using the fixed buffer, the initial processing and the connection open processing must be completed.

(2) The contents of the parameter settings have been loaded into the Ethernet module when the Ethernet module open completion signal (address: 5000_H … corresponding bit) switches from off to on.

(3) The data length in word count is designated (stored) using the dedicated instruction for the communication with the procedure exist control method.
If the send data length exceeds this range at data sending, a communication error occurs and the data is not sent.

(4) Perform the fixed buffer communication using the following dedicated instructions.
- Data sending : BUFSND instruction
- Data receiving : BUFRCV instruction (for main program)
: BUFRCVS instruction (for interrupt program)

For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

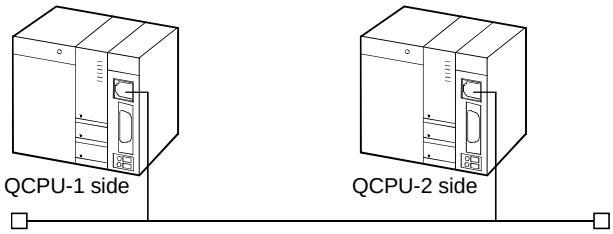
(5) The following should be observed when using a connection opened by UDP.
- External devices can be switched by modifying the setting values in the communication address setting area of the communication parameter setting area before sending/receiving data. Thus, data can be sent to multiple external devices sequentially. When performing sending/receiving, make sure to perform switching between external devices properly so that no communication problems occur.

(6) When reading of the data received from the same connection, reading of data received by the main program and the reading of data received by the interrupt program cannot be done together.
Use only one of the above-mentioned programs for reading the data received.
* Reading of data received by the main program cannot be performed when setting is being performed by GX Developer for reading of received data using the interrupt program.

(7) For data (commnd) transmission, the next data (commnd) should be after the completion of data commnication (such as after the reception of a response) for the transmission of the previous data (commnd).

7.5.2 Fixed buffer communication program example (with the procedure exist control method)

This section explains the programming method in order to communicate data (procedure exist control method) with an external device using the fixed buffers.



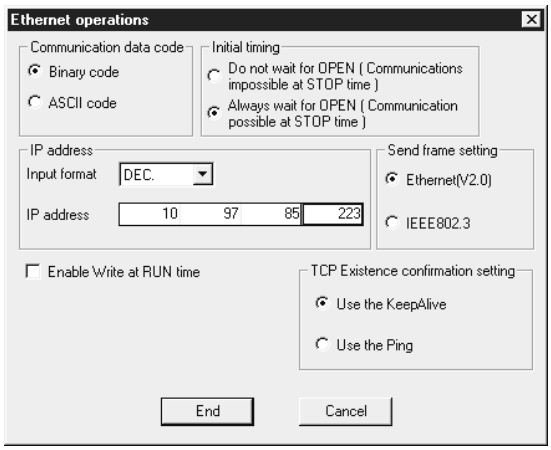
(1) Execution environment for the program example

(a) Send program (QCPU-1st station side)

1) Connection No. 1 is used for fixed buffer sending.
2) The communication parameter settings are assumed to have been set as described in Section 5.6.1, "Active open processing/close processing".
3) Fixed buffer No. 1 send data
   : Stored in D300 to D303
4) Fixed buffer No. 1 send instruction complete device
   : M300
5) Fixed buffer No. 1 send instruction abnormal complete device
   : M301
6) Fixed buffer No. 1 send instruction complete status
   : D3001

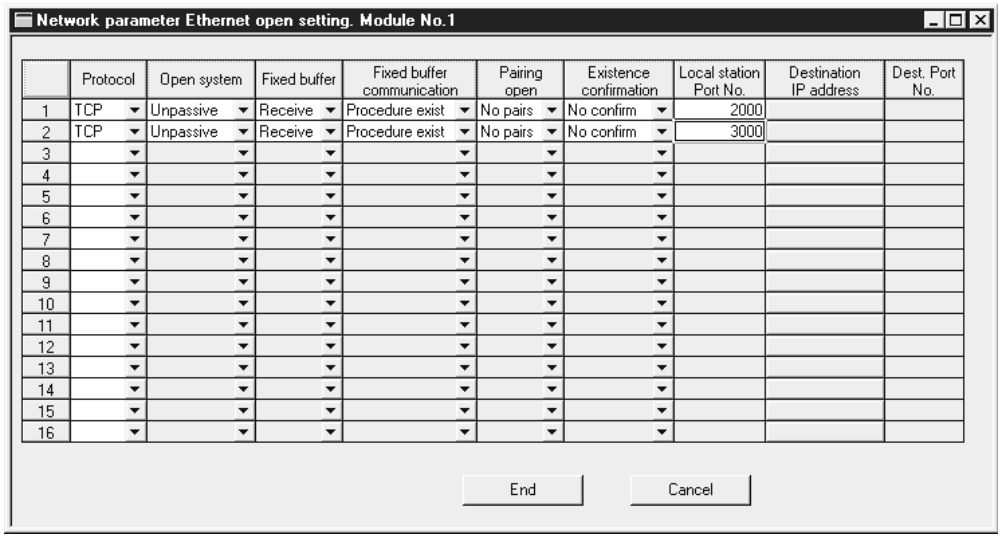
(b) Receive program (QCPU-2nd station side)

1) Connection No. 1 is used for processing the fixed buffer receiving in the main program.
2) Connection No. 2 is used for processing the fixed buffer receiving in the interrupt program.
3) The Ethernet module is mounted in slot "0" of the basic base unit.
4) The "Network parameters Setting the number of MNET/10H Ethernet cards" parameter is assumed to have been set with GX Developer as follows. (See Section 4.6)
   - Network type : Ethernet
   - Head I/O No. : 0000
   - Network No. : 1
   - Group No. : 1
   - Station No. : 2
5) The "Operational settings" parameters are assumed to have been set with GX Developer as follows.



Local station IP address : 0A.61.55.DFH (10.97.85.223)

6) The "Open settings" parameters are assumed to have been set with GX Developer as follows.



Connection No. 1 Local station Port No. : 2000H
(for main program)
Connection No. 2 Local station Port No. : 3000H
(for interrupt program)

7) Fixed buffer No. 1 receive data
: Stored in D500 to D503

8) Fixed buffer No. 2 receive data
: Stored in D700 to D703

9) Fixed buffer No. 1 receive instruction complete device
: M500

10) Fixed buffer No. 1 receive instruction abnormal complete device
: M501

11) Fixed buffer No. 1 send instruction complete status
: D5001

12) Fixed buffer No. 1 receive status signal
: M40

13) The parameter settings to start up the interrupt program are assumed to have been set with GX Developer as described in the screen settings in Section 7.3.2 "Receive processing with an interrupt program".
Fixed buffer No. 2 interrupt pointer
: I50 (SI No. 0)

POINT

Secure sufficient device memory according to the maximum length of data sent from the source in order to prevent device areas used for other purposes from being overwritten by the reception data.

(2) Outline of the program example

(a) Send program (QCPU-1st station side)

1) After setting each parameter with GX Developer and write to PLC CPU, reset the PLC CPU and confirm that the initial processing is completed.

2) Perform open processing (Active open) of connection No. 1. (∗1)

3) Communicate data from the PLC CPU using fixed buffer communication (procedure exist sending).

4) After the data sending is completed, perform close processing of connection No. 1. (∗1)

∗1 Use the program example described in Section 5.6.1, "Active open processing/close processing" for the sequence program that executes the open processing/close processing.

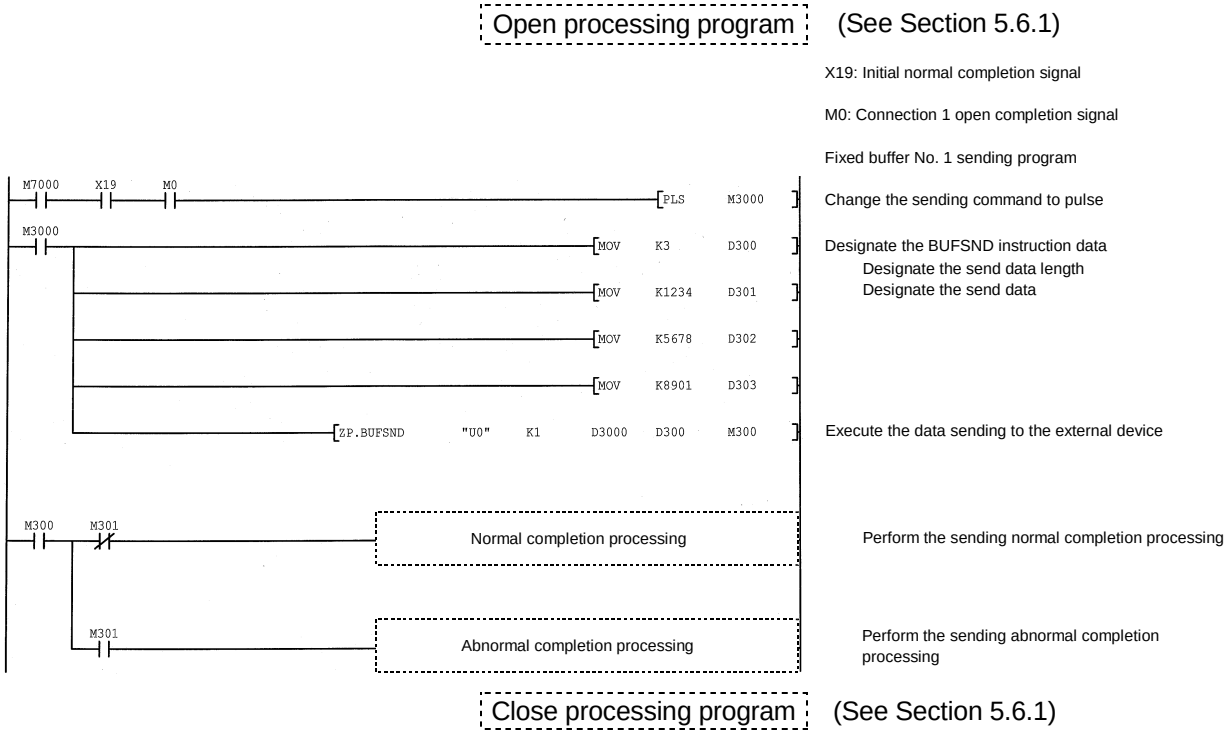
(b) Receive program (QCPU-2nd station side)

1) After setting each parameter with GX Developer and write to PLC CPU, reset the PLC CPU and confirm that the initial processing is completed.
When the initial processing is normally completed, connection No. 1 and 2 wait for an Active open request from the external device.

2) Communicate data from the external device using fixed buffer communication (procedure exist sending).

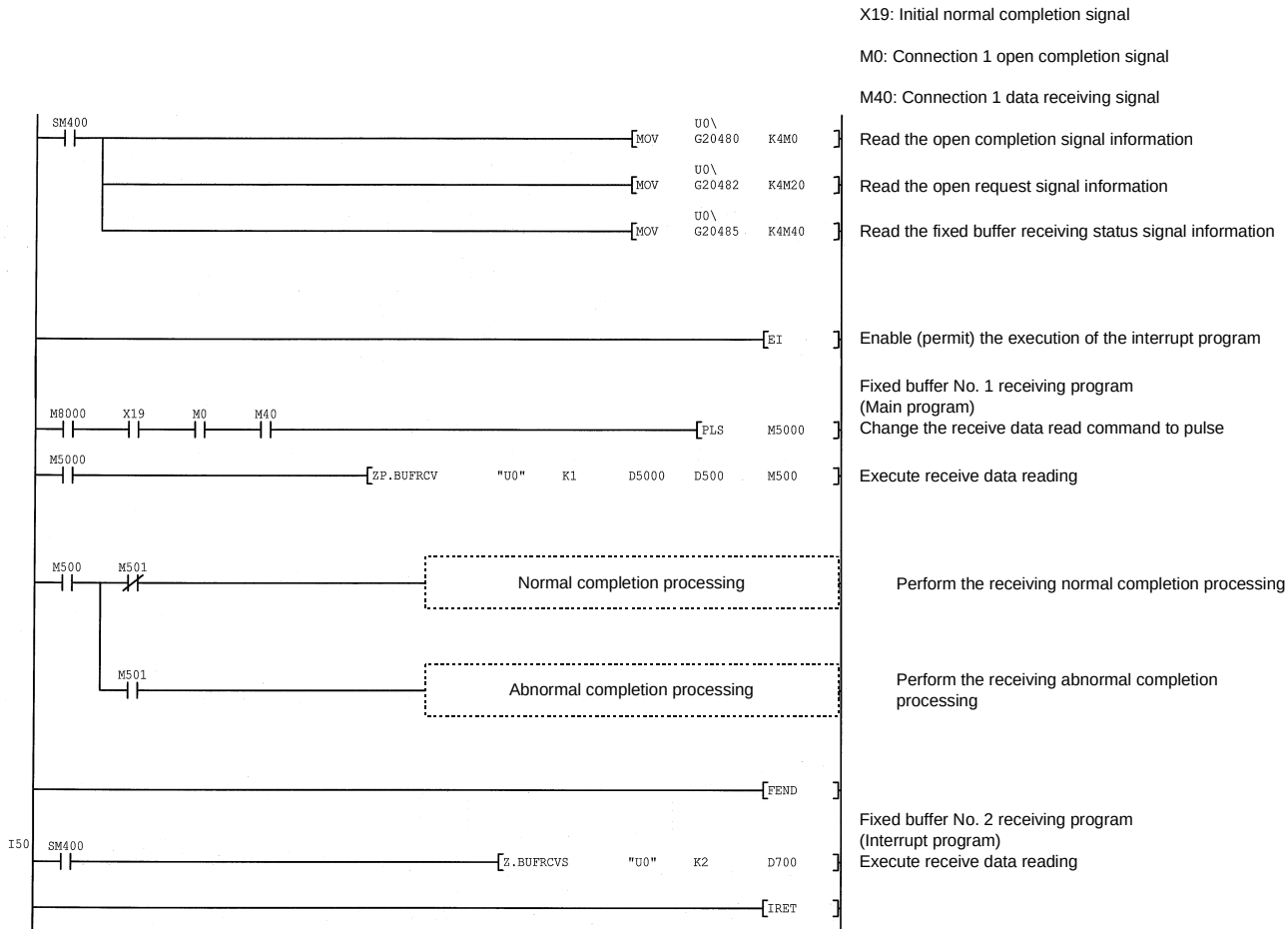
3) Data received in the corresponding fixed buffer data area in the Ethernet module is read to the PLC CPU.

(Send program)



7 - 21

(Receive program)



7 - 22

8 FIXED BUFFER COMMUNICATION (WITH THE NO PROCEDURE CONTROL METHOD)

This chapter explains how the PLC CPU and external device communicate using the fixed buffers (with the no procedure control method) of the Ethernet module.

POINT

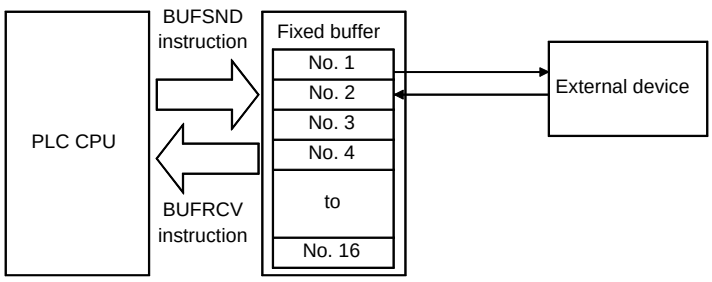
The following points are difference overview from the "Procedure exist" when communicating using the fixed buffers:

1) It is possible to send and receive data which match to the message format of the external device.
   At data sending, subheader, data length, etc. are not included in the application data field of a message; only the data in the fixed buffer is sent. At data receiving, all the data in the message excluding the header is stored in the fixed buffer.
2) A response to data receiving is not sent.
3) Communication is performed using binary code regardless of the communication data that is set with a GX Developer parameter code (see Section 4.7, "Operational Settings".)
4) The maximum application data area is 2046 bytes per communication.
5) The applicable connection is dedicated to the no procedure fixed buffer communication.
   As with the procedure fixed buffer communication, random access buffer communication, and communication using the MC protocol cannot be performed at the same time as the no procedure fixed buffer communication.

8.1 Control Method

The following explains how communication is performed using the fixed buffers and the no procedure control method.
In the communication processing using the fixed buffers, data transmission from the PLC CPU and the external device is executed using the no procedure control method.

(1) The data flow in the communication processing is as follows.



(2) Data can be communicated with the following external devices.
- Device on the Ethernet to which the Ethernet module is connected.
- Devices connected with the router relay function (see Section 5.3)

As shown in the diagram below, when using each fixed buffer (No. 1 to 16), the destination devices and usage conditions (for sending/receiving, procedure exist/no procedure, etc.) should be set when the connection via the Ethernet module is opened to fix the external device for each buffer.

(a) At TCP/IP communication

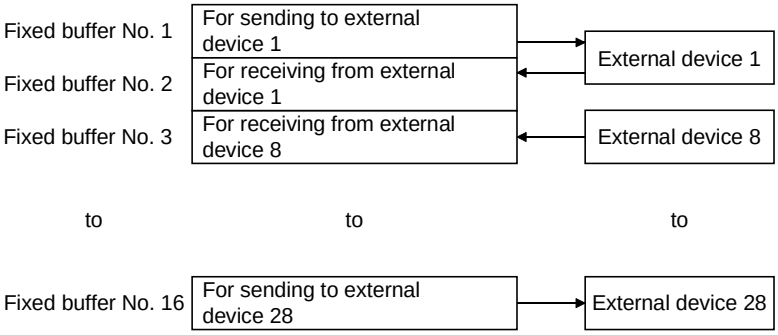
It is allowed to change external devices only when the open completion signal of the applicable connection is off.

(b) At UDP/IP communication

External devices can be changed regardless of the status of the applicable connection.

("Destination IP address" and "Destination Port No." in the communication address setting area can be changed. However, "Local station Port No." cannot be changed.)

When changing external devices, do not use the "Pairing open" and "Existence confirmation" functions.



POINT

The connections for which no procedure is selected are dedicated to the fixed buffer sending or receiving after the completion of open processing.

(3) At data sending/receiving, the Ethernet module performs the following processing.

1) At data sending

When the PLC CPU executes the dedicated BUFSND instruction (*1) in a sequence program, the Ethernet module sends data in the applicable fixed buffer (No. n) to the external device that is specified in the communication address setting area (addresses: 28H to 5FH and 5038H to 507FH) corresponding to fixed buffer No. n. (*2)

2) At data receiving

The Ethernet module processes the receive data if the data is received from an external device set in the communication setting area that corresponds to fixed buffer No. n. (*2)

Also, when the Ethernet module stores the receive data in the corresponding fixed buffer in the receive processing, it updates the destination IP address and destination port No. in the connection information area (addresses: 78H to C7H and 5820H to 586FH) that corresponds to fixed buffer No. n.

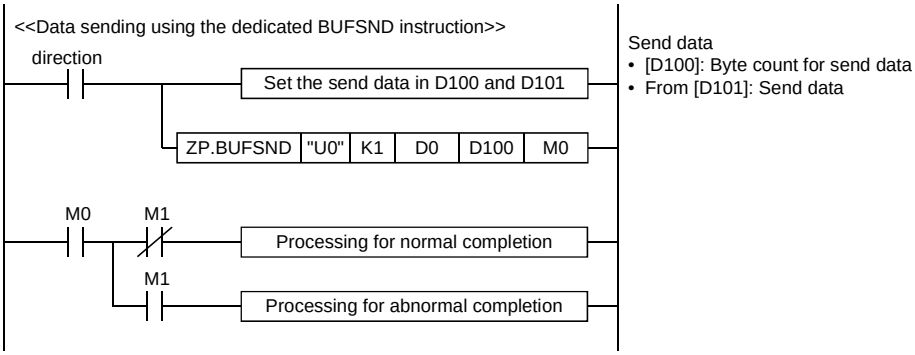
If data is received from an external device not set in the connection information area of the buffer memory, the Ethernet module ignores the receive data to.

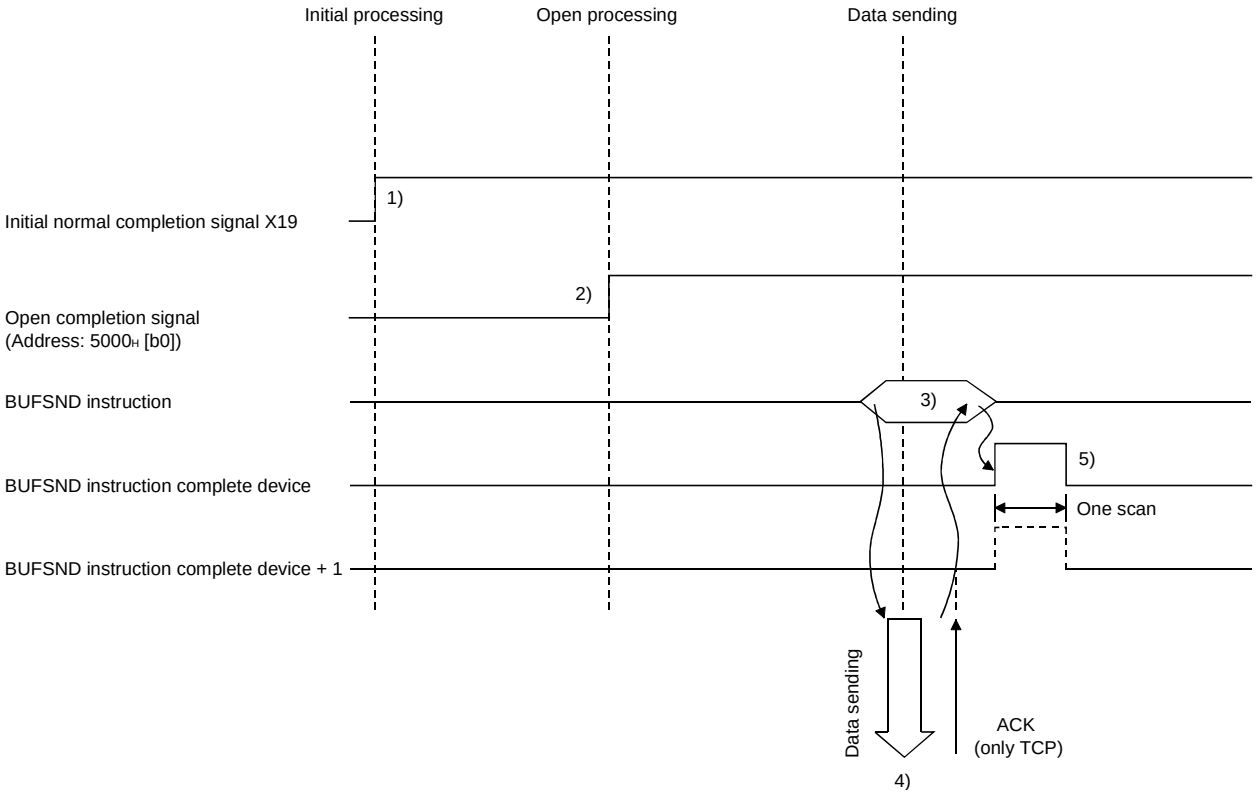
*1 For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

*2 In case of TCP/IP Unpassive open, data is communicated with an external device stored in the connection information area of the buffer memory.

8.2 Sending Control Method

This section explains the control method when data is sent from the Ethernet module to an external device using fixed buffer No. 1 and the area corresponding to connection No. 1 as an example.





1) Confirm normal completion of the initial processing.

2) Confirm the normal completion of the open processing for connection No. 1.

3) Execute the dedicated BUFSND instruction.
The Ethernet module processes the following and sends the data.
• Writes send data length and send data to the fixed buffer (No. 1) area.
Send data length : The head address area of the target fixed address (*1)
Send data : Area beginning from the head address of the target fixed buffer + 1
*1 The send data length is expressed by a byte count.

4) Only the size of the send data in the fixed buffer (No. 1) designated by the send data length is sent to the designated external device (set in the open processing).

5) The Ethernet module terminates the data transmission.

At normal completion
• BUFSND instruction complete device : ON
• BUFSND instruction complete device + 1 : OFF
• BUFSND instruction complete status area (*2) : 0000H

At abnormal completion :
• BUFSND instruction complete device : ON
• BUFSND instruction complete device + 1 : ON
• BUFSND instruction complete status area (*2) : Value other than 0000H

After the data transmission is abnormally completed, execute the dedicated BUFSND instruction again to repeat the transmission processing.

*2 The status at completion is stored in the complete status area of the dedicated instructions. For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

POINT

The following precaution should be observed when communicating using UDP/IP:
• When the Ethernet module's internal processing is normally completed, data send processing ends even if the communication line between the PLC CPU and an external device is disconnected because of cable disconnection, etc. It is recommended to send/receive data using a communication procedure defined by the user.

8.3 Receiving Control Method

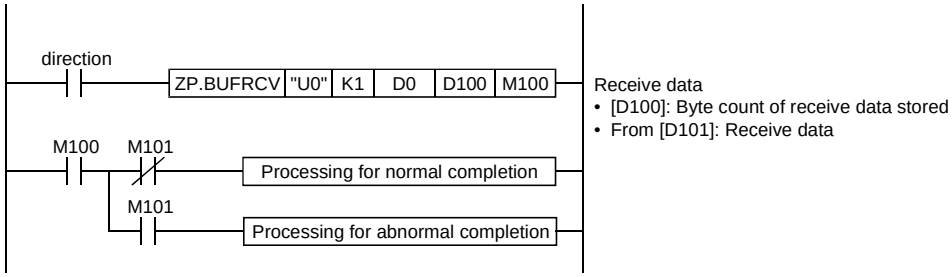
This section explains the control method when the Ethernet module receives data from an external device.
Fixed buffer communication employs the following receive processing methods:

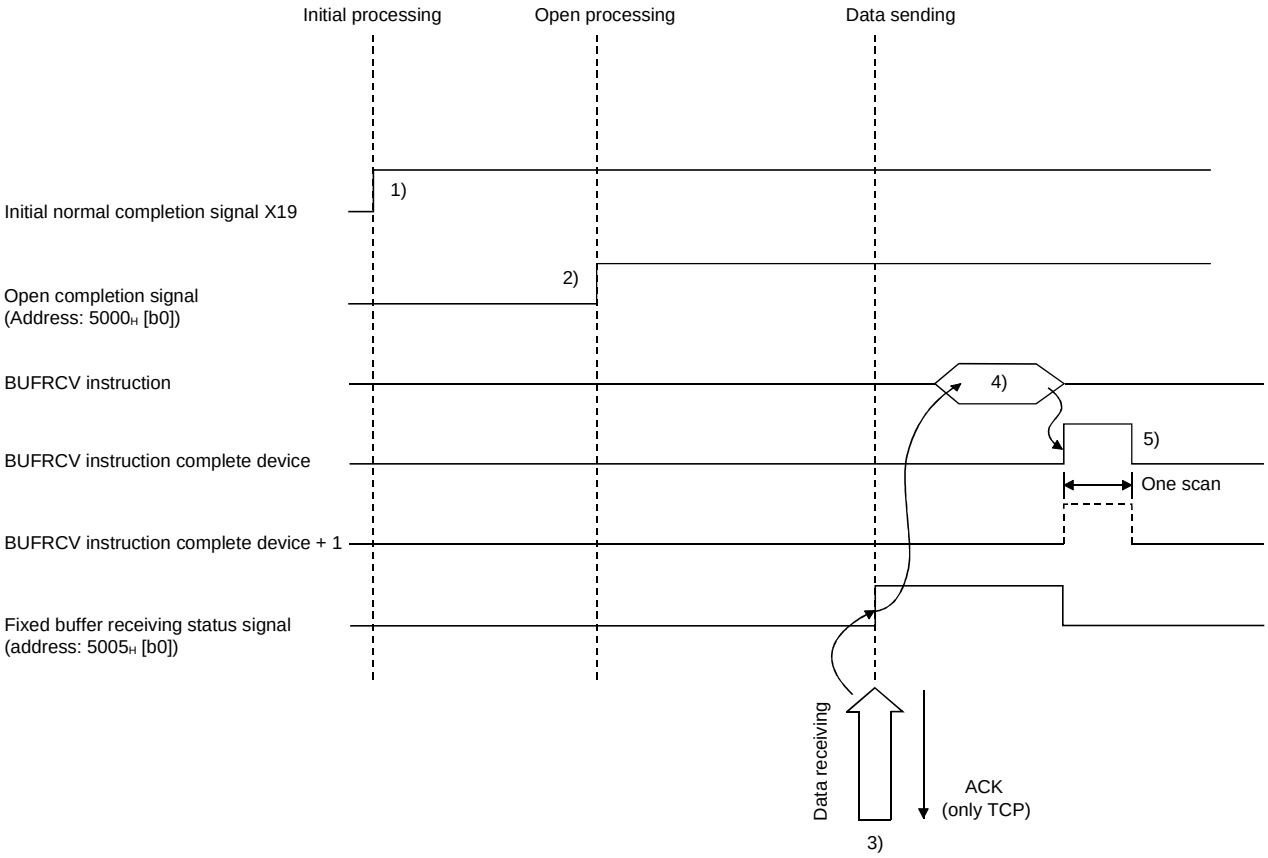
- Receive processing with the main program (dedicated instruction: BUFRCV)
  : See Section 8.3.1
- Receive processing with an interrupt program (dedicated instruction: BUFRCVS)
  : See Section 8.3.2

8.3.1 Receive processing with the main program (dedicated instruction: BUFRCV)

This section explains about the receive processing to be performed with the main program, using an example in which the fixed buffer No. 1 and the area corresponding to connection No. 1.

<<Data receiving using the dedicated BUFRCV instruction (main program)>>





1) Confirm the normal completion of the initial processing.
2) Confirm the normal completion of the open processing for connection No. 1.
3) Upon receiving data from the designated external device (set in the open processing), the Ethernet module processes the following.
   - Stores the receive data to the fixed buffer (No. 1) area. (Area beginning from the head address of the target fixed buffer + 1)
   - Stores the data length to the head address area of the target fixed address (*1)
   - Fixed buffer receive status signal (address: 5005H … b0) : ON

   *1 The receive data length is expressed by a byte count. When an odd number of data bytes is received, the last received data is stored at the lower byte of the last data storage area. (The higher byte becomes a non-constant value.)
4) Execute the dedicated BUFRCV instruction and read the receive data length and receive data from the fixed buffer (No. 1).
5) End the receive processing.

   At normal completion
   - BUFRCV instruction complete device : ON
   - BUFRCV instruction complete device + 1 : OFF
   - BUFRCV instruction complete status area (*2) : 0000H

   At abnormal completion
   - BUFRCV instruction complete device : ON
   - BUFRCV instruction complete device + 1 : ON
   - BUFRCV instruction complete status area (*2) : Value other than 0000H

   *2 The status at completion is stored in the complete status area of the dedicated instructions. For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

POINT

(1) The destination setting (see Section 5.5) for a connection whose parameters are set with GX Developer becomes valid when the open completion signal (address: 5000H … corresponding bit) of the Ethernet module switches from off to on.

(2) Execute the BUFRCV instruction when the corresponding connection's bit in the fixed buffer receive status signal storage area (address: 5005H) of the buffer memory switches from off to on.

(3) At abnormal data receiving, the fixed buffer receive completion signal (address: 5005H … b0) does not turn on.
In addition, data is not stored in the fixed buffer (No. 1) area.

8.3.2 Receive processing with an interrupt program (dedicated instruction: BUFRCVS)

This section explains about the receive processing when an interrupt program is used. When an interrupt program is set up to handle the receive processing, the interrupt program starts up when data is received from an external device and reading the receive data destined for the PLC CPU is enabled.

In order to start up the interrupt program, set the parameters using GX Developer.

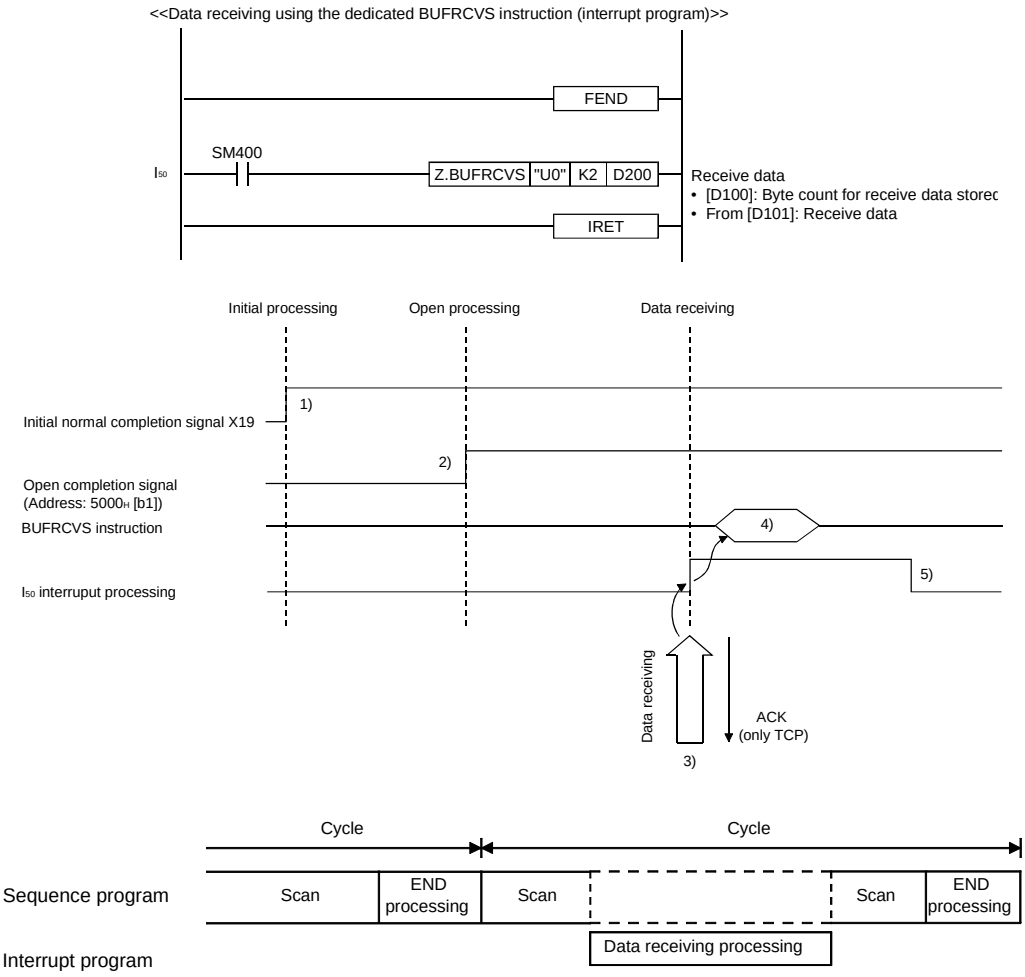
(1) Setting screen

Set the following parameters using GX Developer to start the interrupt program. The setting screen and setting method are the same as for the fixed buffer communication (procedure exist). See Section 7.3.2 (1).

(2) Control method

The control method when the interrupt sequence program starts up is explained using fixed buffer No. 2 and the area corresponding to connection No. 2 as an example.

* The interrupt settings with GX Developer for reading the receive data using the interrupt program example below is illustrated in the screen display shown in Section 7.3.2 (1).



1) Confirm the normal completion of the initial processing.

2) Confirm the normal completion of the open processing for connection No. 2.

3) Upon receiving data from the designated external device (set in the open processing), the Ethernet module processes the following.
   - Stores the receive data in the fixed buffer (No. 2) area. (Area beginning from the head address of the target fixed buffer + 1)
   - Stores the data length in the head address area of the target fixed address (∗1)
   - Request the PLC CPU to start up the interrupt program.

   ∗1 The receive data length is expressed by a byte count. When an odd number of data bytes is received, the last received data is stored at the lower byte of the last data storage area. (The higher byte becomes a non-constant value.)

4) The interrupt program starts up.
   Execute the dedicated BUFRCVS instruction and read the receive data length and receive data from the fixed buffer (No. 2).

   At normal completion
   - PLC CPU error flag (SMO) (∗2) : OFF

   At abnormal completion
   - PLC CPU error flag (SMO) (∗2) : ON
   - PLC CPU error code (SDO) (∗2) : error code

   ∗2 For more details on the PLC CPU error flag (SMO) and error code (SDO), see the manual for the PLC CPU used.

5) The execution of the interrupt program is finished and the execution of the main program starts again.

REMARK

(1) To start up an interrupt program, create an interrupt enable/disable program in the main program using the EI and DI instructions.

(2) The programming for reading of receive data with an interrupt program is the same as that for communicating (procedure exist) by fixed buffer. Refer to program example indicated in Section 7.5.2 when programming.

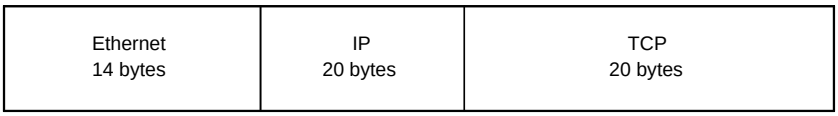
8.4 Data Format

When communicating data between the Ethernet module and an external device, the data format explained below is used.
The communication data consists of a "header" and "application data" as shown below.

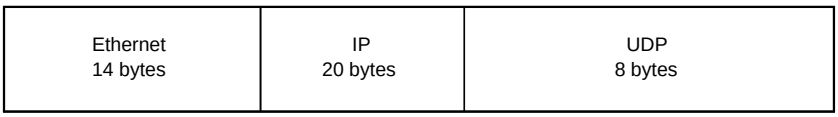
(1) Header

The header for TCP/IP or UDP/IP is used. In case of the Ethernet module, the Ethernet module adds and deletes the header. Thus, the user does not need to set it.
(Details of the size of the header section)

1) In case of TCP/IP

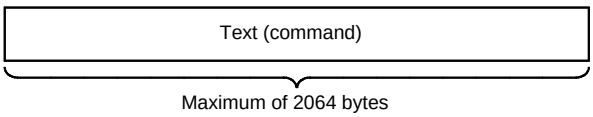


2) In case of UDP/IP



(2) Application Data

The data code in the application data is expressed in binary code.
Communication is performed using binary code, regardless of the set communication data (see Section 4.7).



REMARK

The subheader and data length that are added for communications using the fixed buffers in the procedure exist control method are not present for communications in the no procedure control method. All data is treated as valid text.

8.5 Simultaneous Broadcast Using UDP/IP

When UDP/IP is used for no procedure fixed buffer communication, data can be broadcast simultaneously to all Ethernet module installed stations on the same Ethernet where the Ethernet module is located. This allows writing the same data to all stations, etc.

POINT

(1) In simultaneous broadcasting, if the messages received by the simultaneous broadcasting are not needed, external devices connected to the same Ethernet should discard them at first given opportunity.

(2) The user needs to determine the port numbers dedicated to the transmission and reception by simultaneous broadcasting, after which simultaneous broadcasting can be performed by designating these port numbers.

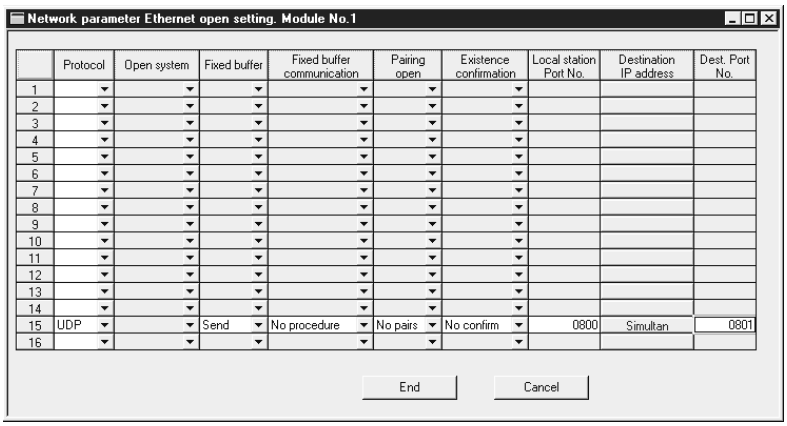
8.5.1 Sending by simultaneous broadcasting

Sending can be performed through simultaneous broadcasting by setting the external device IP address to FFFFFFFFH and performing the open processing.
For data sending by simultaneous broadcasting, the Ethernet sets the request destination IP address to FFFFFFFFH and send data over the Ethernet.

(1) Setting screen

This section explains the "Open settings" for sending by simultaneous broadcasting.

- Connection No. 15 is used.
- The Ethernet module Port No. is 0800H.



(a) Protocol
Select "UDP/IP".

(b) Open system
Setting is not required.

(c) Fixed buffer
Select "Send".

(d) Fixed buffer communication
Select "No procedure".

(e) Pairing open
Select "No pairs".

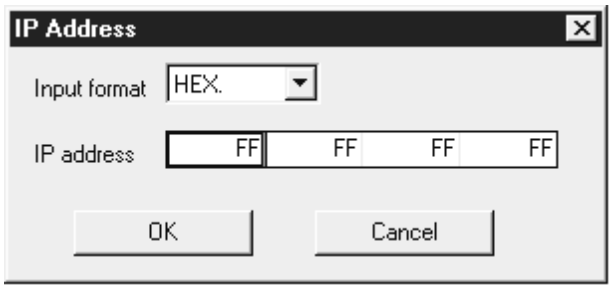
(f) Existence confirmation
Select "No confirm".

(g) Local station Port No.

1) Set the Ethernet module port No. in hexadecimal.

2) Designate the setting value in a range from 401H to 1387H and 138BH to FFFEH upon consulting a network administrator. Select a port No. not used by other ports.

(h) Destination IP address
Set "FFFFFFFFH".



(i) Destination Port No.

1) Set the port number of the external device side in hexadecimal.

2) Designate the setting value in a range from 401H to FFFEH upon consulting a network administrator.

3) Check the communication range shown in Section 8.1 (2).

(2) Send control method

The send control method is the same as for the normal no-procedure communication using the fixed buffers. For more details, see Section 8.2.

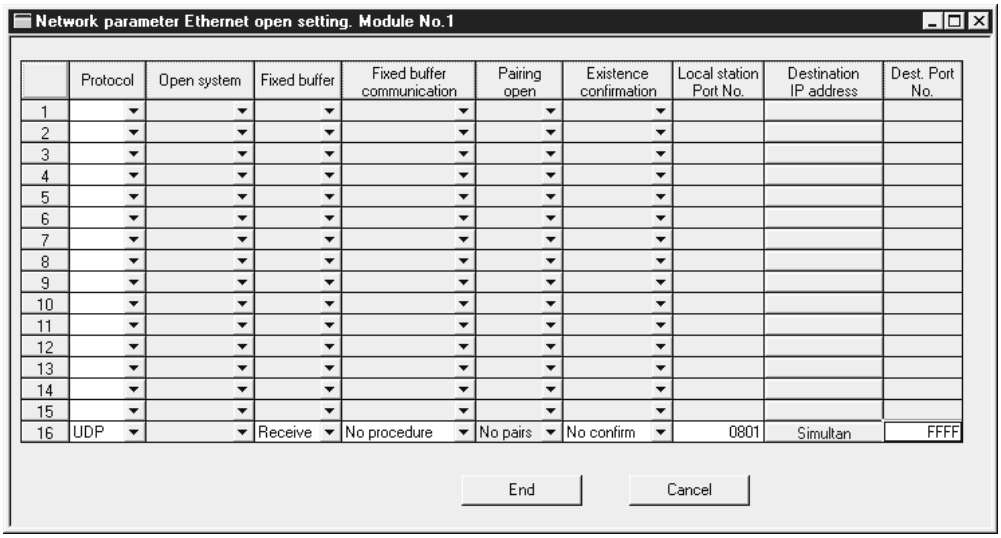
8.5.2 Receiving by simultaneous broadcasting

Data is received through simultaneous broadcasting, by setting FFFFFFFFH for the IP address of an external device that is to send data to the Ethernet module, and FFFFH for the port No., then performing the open processing.

(1) Setting screen

This section explains the "Open settings" for receiving by simultaneous broadcasting.

- Connection No. 16 is used.
- The Ethernet module Port No. is 0801H.



(a) Protocol
Select "UDP/IP".

(b) Open system
Setting is not required.

(c) Fixed buffer
Select "Receive".

(d) Fixed buffer communication
Select "No procedure".

(e) Pairing open
Select "No pairs".

(f) Existence confirmation
Select "No confirm".

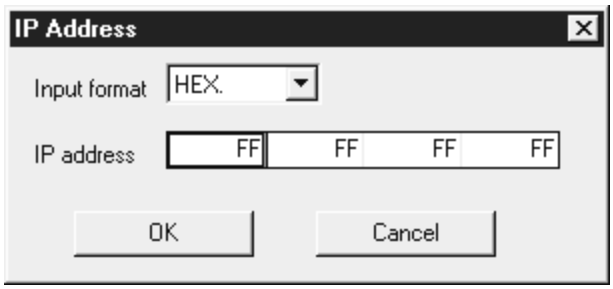
(g) Local station Port No.

1) Set the Ethernet module port No. in hexadecimal.

2) Designate the setting value in a range from 401H to 1387H and 138BH to FFFEH.

3) If the receive data from specific external device is to be processed, designate the user-defined port No. upon consulting a network administrator.

(h) Destination IP address
Set "FFFFFFFFH".



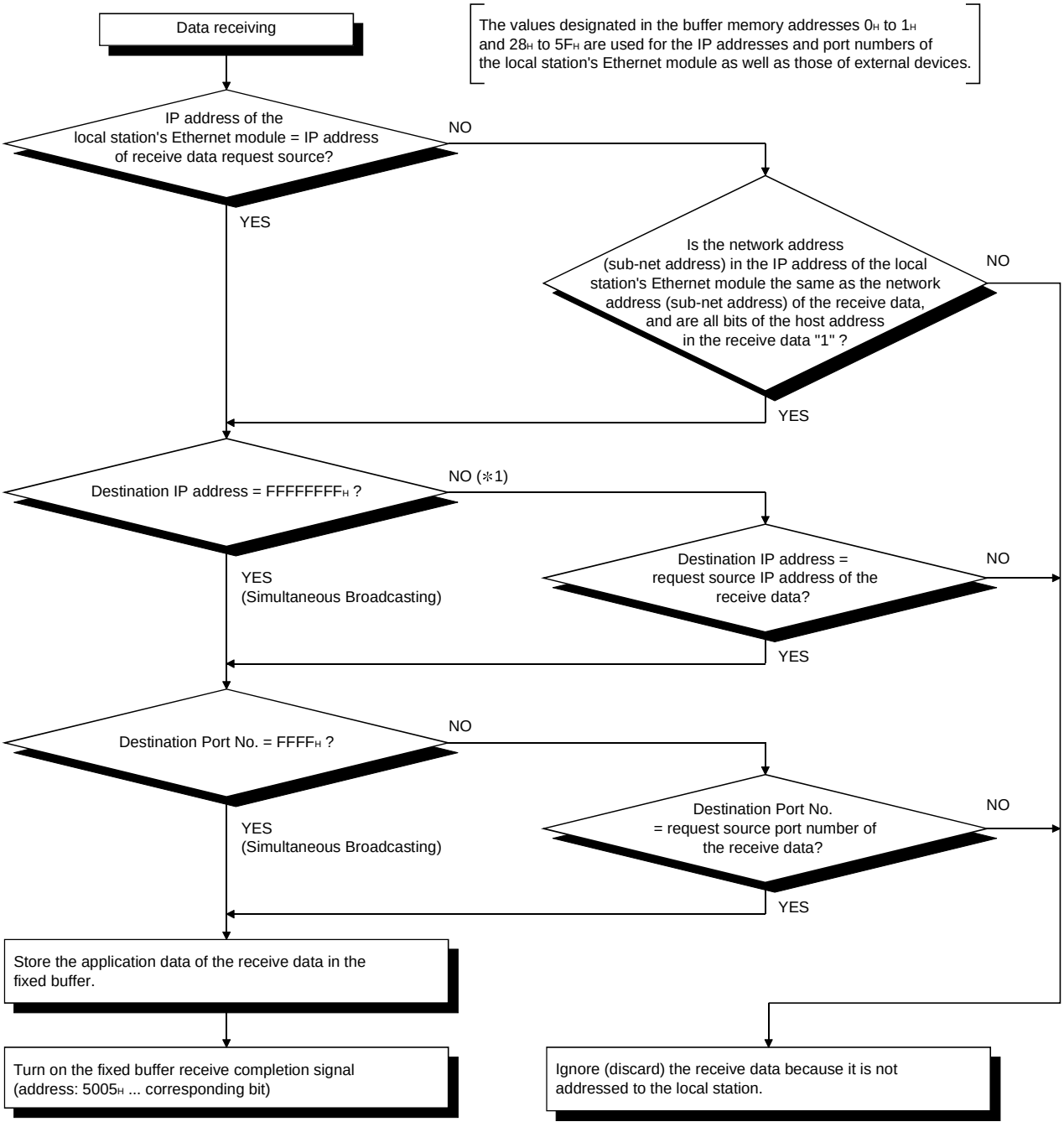
(i) Destination Port No.
Set "FFFFH".

(2) Receive control method

The receive control method is the same as for the normal no procedure communication using the fixed buffers. For more details, see Section 8.3.

REMARK

The following flowchart illustrates an outline of the Ethernet module's internal processing when data is received by simultaneous broadcasting in the no procedure control method.


Data receiving
The values designated in the buffer memory addresses 0H to 1H and 28H to 5FH are used for the IP addresses and port numbers of the local station's Ethernet module as well as those of external devices.
IP address of the local station's Ethernet module = IP address of receive data request source?
NO
YES
Is the network address (sub-net address) in the IP address of the local station's Ethernet module the same as the network address (sub-net address) of the receive data, and are all bits of the host address in the receive data "1" ?
NO
YES
Destination IP address = FFFFFFFFH ?
NO (*1)
YES (Simultaneous Broadcasting)
Destination IP address = request source IP address of the receive data?
NO
YES
Destination Port No. = FFFFH ?
NO
YES (Simultaneous Broadcasting)
Destination Port No. = request source port number of the receive data?
NO
YES
Store the application data of the receive data in the fixed buffer.
Turn on the fixed buffer receive completion signal (address: 5005H ... corresponding bit)
Ignore (discard) the receive data because it is not addressed to the local station.

*1 If the bits in the range that indicates the host address of the receive data request destination IP address are all "1's," the processing is performed by the YES side.

8.5.3 Precautions when using the simultaneous broadcast function

The following precautions should be observed when broadcasting simultaneously in the no procedure communication using the fixed buffers.

(1) The user needs to determine the port No. dedicated to the sending and receiving by simultaneous broadcasting, after which simultaneous broadcasting can be performed by designating these port No..

(2) The send message by simultaneous broadcasting is sent to all external devices on the same Ethernet to which the Ethernet module is connected.
If the messages received by simultaneous broadcasting are not needed, the external devices that are connected to the same Ethernet should discard them.

* The external devices should discard the message received if it is not needed. Also, do not return a response even if the external device is the station to which the message received is addressed. The Ethernet module performs these processes automatically.

(3) A maximum of 2046 bytes of data in the application data field can be processed per sending/receiving.
When it is necessary to send/receive data that exceeds 2047 bytes, divide it into smaller portions at the send source.

(4) When performing simultaneous broadcasting, set the "Existence confirmation" setting to "No confirm" in the "Open settings" of the corresponding connection.

REMARK

The size of the internal buffer for the operating system is approximately 40 k bytes and any received data that exceeds this size will be discarded.

8.6 Programming

This section explains the programming method when the Ethernet module and an external device communicate using the fixed buffers and the no-procedure control method.

8.6.1 Precautions when creating programs

(1) In order to communicate using the fixed buffers, the initial processing and the connection open processing must be completed.

(2) The contents of the parameter settings have been loaded into the Ethernet module when the Ethernet module open completion signal (address: 5000H... corresponding bit) switches from off to on.

(3) The data length in byte count is designated (stored) using the dedicated instruction for the no procedure communication.
If the send data length exceeds this range at data sending, a communication error occurs and the data is not be sent.

(4) Perform the fixed buffer communication using the following dedicated instructions.
- Data sending : BUFSND instruction
- Data receiving : BUFRCV instruction (for main program)
: BUFRCVS instruction (for interrupt program)

For details on the dedicated instructions, see Chapter 10, "Dedicated Instructions".

(5) The following should be observed when using a connection opened by UDP.
- External devices can be switched by modifying the setting values in the communication address setting area of the communication parameter setting area before sending/receiving data. Thus, data can be sent to multiple external devices sequentially. When performing sending/receiving, make sure to perform switching between external devices properly so that no communication problems occur.

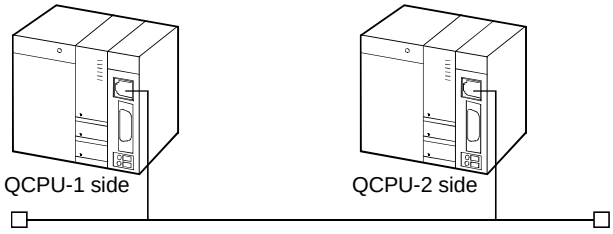
(6) The connections for which no procedure is selected are dedicated to the no procedure fixed buffer sending/receiving. Thus, communications using fixed buffers with the procedure exist control method, random access buffers and the MC protocol cannot be performed at the same time with a communication using fixed buffers with the no procedure control method.

(7) Message data length is not used for the no procedure communication.
The Ethernet module stores the size of the received message (packet) in the receive data length storage area and turns on the fixed buffer receiving status signal (address: 5005H … corresponding bit).
It is recommended to employ a check system, such as including the data length and data type code in the application data of a message, so that the byte count and data type in the application data can be identified by the receiving side.

(8) When reading of the data received from the same connection, reading of data received by the main program and the reading of data received by the interrupt program cannot be done together.
Use only one of the above-mentioned programs for reading the data received.
* Reading of data received by the main program cannot be performed when setting is being performed by GX Developer for reading of received data using the interrupt program.

8.6.2 Fixed buffer communication program example (with the no procedure control method)

This section explains the programming method in order to communication data (through the no procedure control method) with an external device using the fixed buffers.



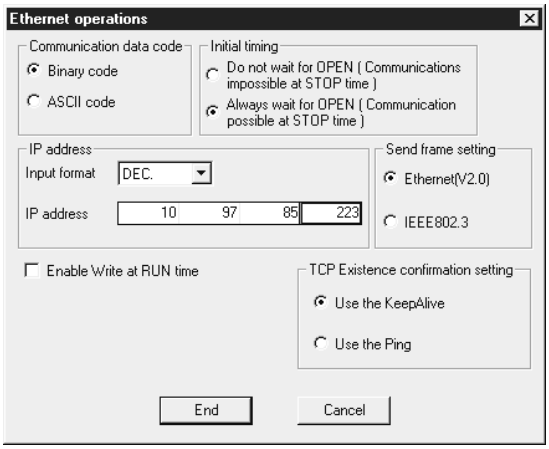
(1) Execution environment for the program example

(a) Send program (QCPU-1st station side)

1) Connection No. 1 is used for fixed buffer sending.
2) The communication parameters settings are assumed to have been set as described in Section 5.6.1, "Active open processing/close processing" except the "Fixed buffer communication" parameter. The "Fixed buffer communication" setting should be changed from "Procedure exist" to "No procedure".
3) Fixed buffer No. 1 send data
: Stored in D300 to D303
4) Fixed buffer No. 1 send instruction complete device
: M300
5) Fixed buffer No. 1 send instruction abnormal complete device
: M301
6) Fixed buffer No. 1 send instruction complete status
: D3001

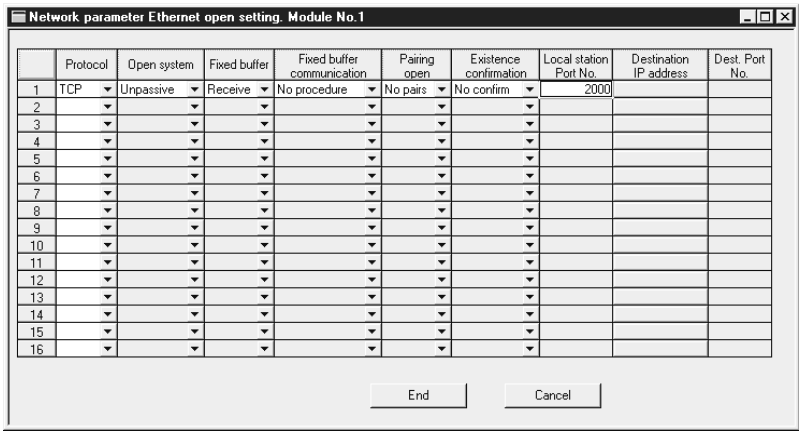
(b) Receive program (QCPU-2nd station side)

1) Connection No. 1 is used for processing the fixed buffer receiving.
2) The Ethernet module is loaded in slot "0" of the basic base unit.
3) The "Network parameters Setting the number of MNET/10H Ethernet cards" parameters are assumed to have been set with GX Developer as follows. (See Section 4.6.)
   - Network type : Ethernet
   - Head I/O No. : 0000
   - Network No. : 1
   - Group No. : 1
   - Station No. : 2
4) The "Operational settings" parameters are assumed to have been set with GX Developer as follows.



Local station IP address: 0A.61.55.DFH (10.97.85.223)

5) The "Open settings" parameters are assumed to have been set with GX Developer as follows.



Connection No. 1 Local station Port No.: 2000H (for main program)

6) Fixed buffer No. 1 receive data
: Stored in D500 to D503

7) Fixed buffer No. 1 receive instruction complete device
: M500

8) Fixed buffer No. 1 receive instruction abnormal complete device
: M501

9) Fixed buffer No. 1 send instruction complete status
: D5001

10) Fixed buffer No. 1 receive status signal
: M40

POINT
Secure sufficient device memory according to the maximum length of data sent from the source in order to prevent device areas used for other purposes from being overwritten by the reception data.

(2) Outline of the program example

(a) Send program (QCPU-1st station side)

1) After setting each parameter with GX Developer and write to PLC CPU, reset the PLC CPU and confirm that the initial processing is completed.

2) Perform open processing (Active open) for connection No. 1. (*1)

3) Communicate data from the PLC CPU using fixed buffer communication (no procedure sending).

4) After the data sending is completed, perform close processing for connection No. 1. (*1)

*1 Use the program example described in Section 5.6.1, "Active open processing/close processing" for the sequence program that executes the open processing/close processing.
Make sure to change the "Fixed buffer communication" setting from "Procedure exist" to "No procedure".

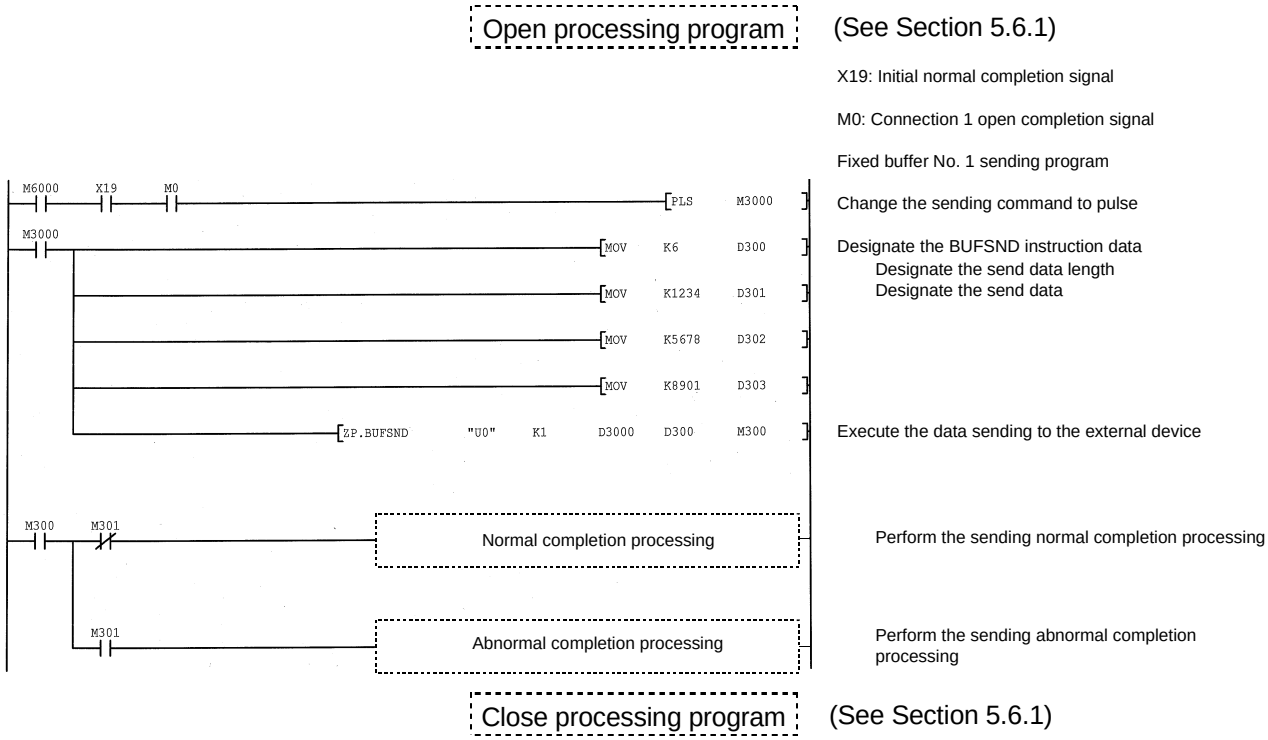
(b) Receive program (QCPU-2nd station side)

1) After setting each parameter with GX Developer and write to PLC CPU, reset the PLC CPU and confirm that the initial processing is completed.
When the initial processing is normally completed, connection No. 1 waits for an Active open request from the external device.

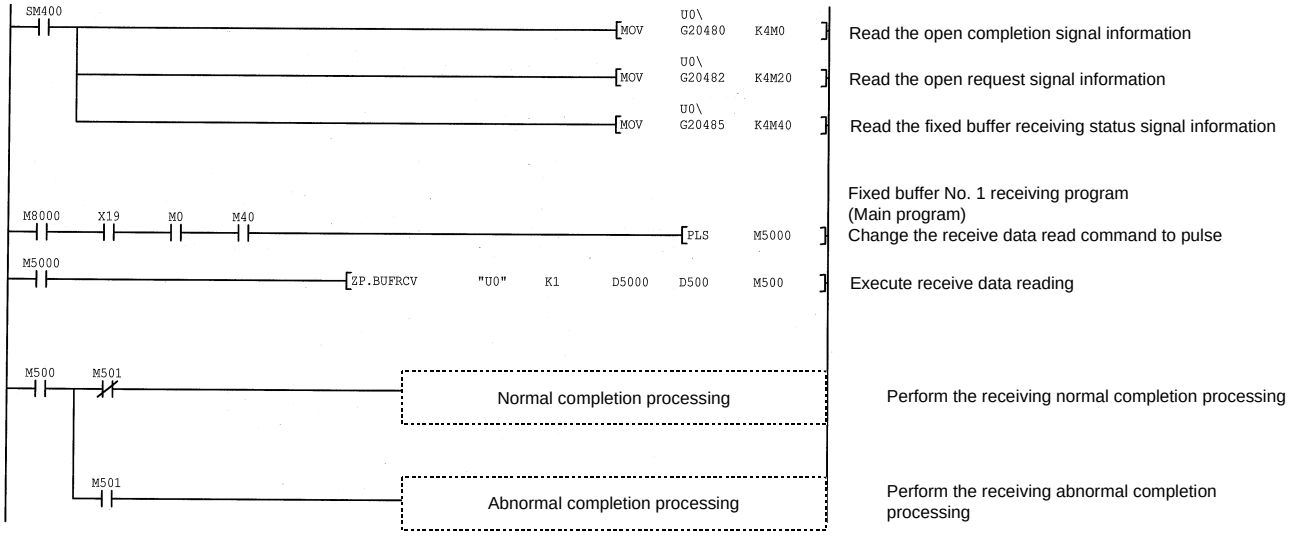
2) Communicate data from the external device using fixed buffer communication (no procedure sending).

3) Data received in the corresponding fixed buffer data area in the Ethernet module is read to the PLC CPU.

(Send program)



(Receive program)


X19: Initial normal completion signal
M0: Connection 1 open completion signal
M40: Connection 1 data receiving signal
SM400
MOV U0\G20480 K4M0
Read the open completion signal information
MOV U0\G20482 K4M20
Read the open request signal information
MOV U0\G20485 K4M40
Read the fixed buffer receiving status signal information
Fixed buffer No. 1 receiving program
(Main program)
M8000 X19 M0 M40
PLS M5000
Change the receive data read command to pulse
M5000
ZP.BUFRCV "U0" K1 D5000 D500 M500
Execute receive data reading
M500 M501
Normal completion processing
Perform the receiving normal completion processing
M501
Abnormal completion processing
Perform the receiving abnormal completion processing

8 - 21

9 COMMUNICATION USING THE RANDOM ACCESS BUFFER

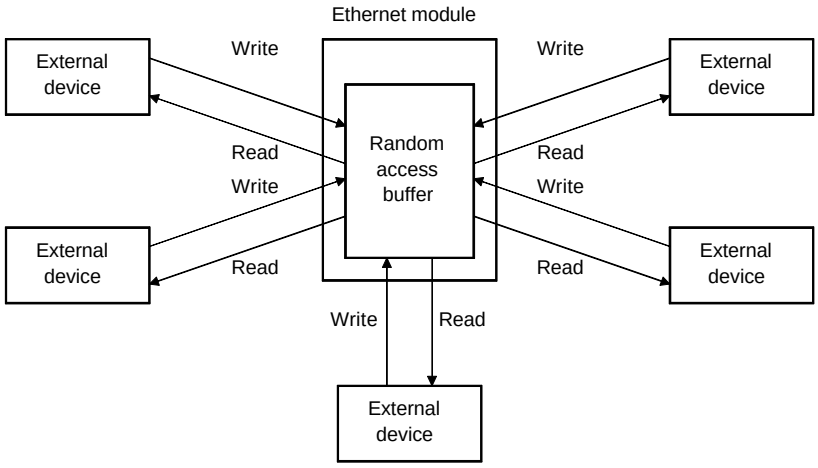
This chapter explains how to communicate data between the Ethernet module and external devices using the random access buffer of the Ethernet module.

POINT
The communication function using the random access buffer and the e-mail sending/receiving function of the PLC CPU cannot be used together. The communication function using the random access buffer and the e-mail sending function of the PLC CPU monitoring function can, however, be used together.

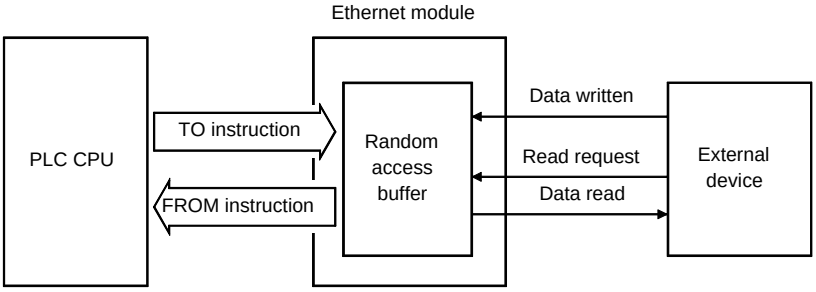
9.1 Control Method

This section explains the control method of random access buffer communication.
In random access buffer communication, data is written to and read from the random access buffer according to instructions (requests) from an external device.
The external device writes and reads data in the random access buffer of the Ethernet module asynchronously with the sequence program of the PLC CPU.

(1) The random access buffer can be written to and read from freely by any external device (excluding the Ethernet module) without giving access to a specific external device.
Thus, it can be used as a common buffer area for all of the external devices connected to the Ethernet.



(2) The following shows the data flow in the communication processing using the random access buffer.



(3) The random access buffer communication can be performed from all external devices except from the Ethernet module itself (including conventional modules). The random access buffer communication cannot be used for communication between PLC CPUs.

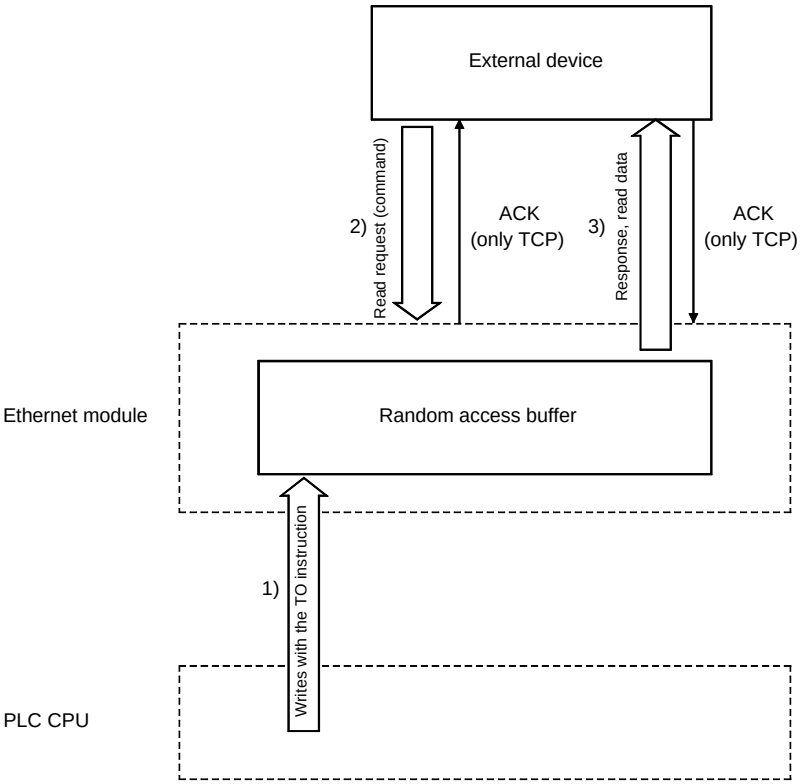
(External devices that can perform communication using the random access buffer)

- External devices on the Ethernet to which the Ethernet module is connected.
- External devices on the Ethernet that are connected with the router relay function (see Section 5.3)

9 - 2

9.1.1 Control method by read requests from an external device

The following diagram illustrates the control method when data is sent from the Ethernet module in response to a read request from the external device.



1) The PLC CPU writes data to the random access buffer of the Ethernet module according to the TO instruction of the sequence program. Or, the external device writes data to the random access buffer of the Ethernet module.

2) Asynchronously with the processing of 1) above, the external device sends a read request to the Ethernet module. (The Ethernet module side: command receiving)

3) Upon receiving the read request from the external device, the Ethernet module sends the data written in the random access buffer to the external device which sent the read request. (The Ethernet module side: response sending)

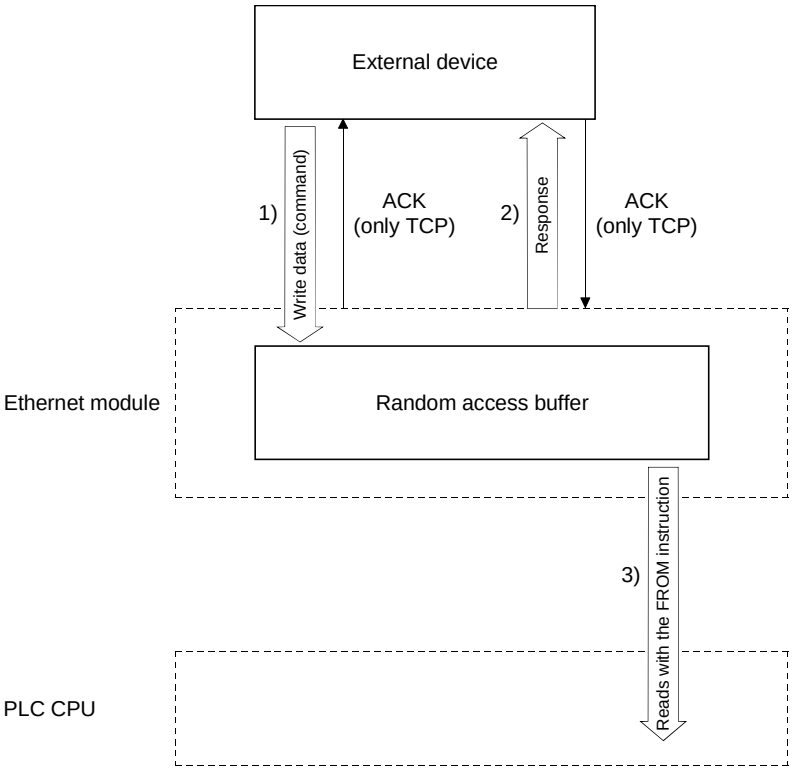
POINT

(1) In random access buffer communication, data can be communicated with an external device for which the procedure exist control method is selected in the fixed buffer communication procedure setting (see Section 5.5) using a connection where the Ethernet open completion signal (address: 5000_H … corresponding bit) is on.

(2) Communication using the random access buffer is performed asynchronously with the sequence program. When it is necessary to synchronize, use the fixed buffer communication function.

9.1.2 Control method by write requests from an external device

The following diagram illustrates the control method when an external device writes data to the random access buffer of the Ethernet module.



1) The external device writes data to the random access buffer of the Ethernet module.
(The Ethernet module side: command receiving)

2) The Ethernet processes the write request from the external device and returns the writing result to the external device that sent the write request.
(The Ethernet module side: response sending)

3) Asynchronously with the processing of 1) and 2) above, the data written to the random access buffer is read with the FROM instruction of the sequence program.

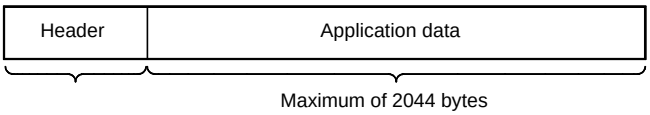
POINT

(1) In random access buffer communication, data can be communicated with an external device for which the procedure exist control method is selected in the fixed buffer communication procedure setting (see Section 5.5) using a connection where the Ethernet open completion signal (address: 5000_H … corresponding bit) is on.

(2) Communication using the random access buffer is performed asynchronously with the sequence program. When it is necessary to synchronize, use the fixed buffer communication function.

9.2 Data Format

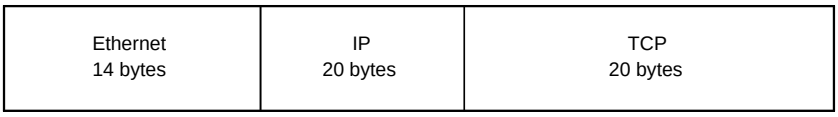
When communicating data between the Ethernet module and an external device, the data format explained below is used.
The communication data consists of a "header" and "application data" as shown below.



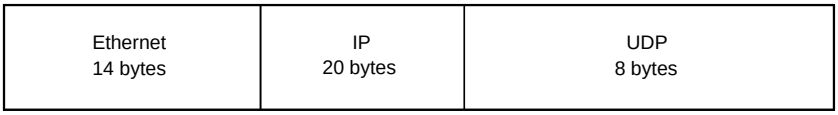
9.2.1 Header

The header for TCP/IP or UDP/IP is used. In case of the Ethernet module, the Ethernet module adds the header.
(Details of the size of the header section)

1) In case of TCP/IP



2) In case of UDP/IP

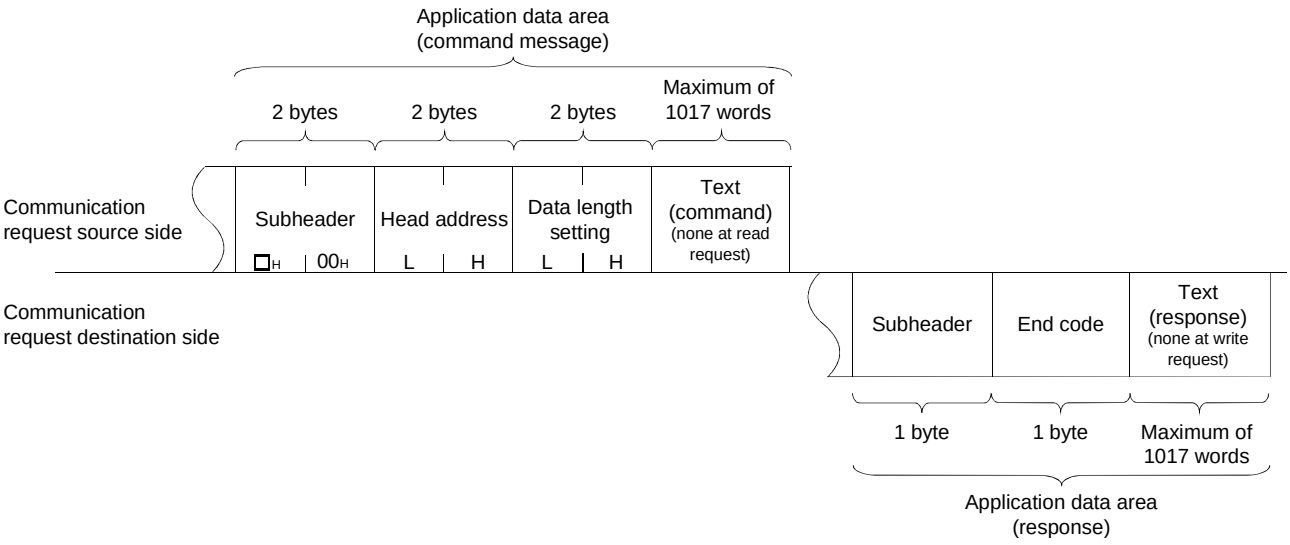


9.2.2 Application data

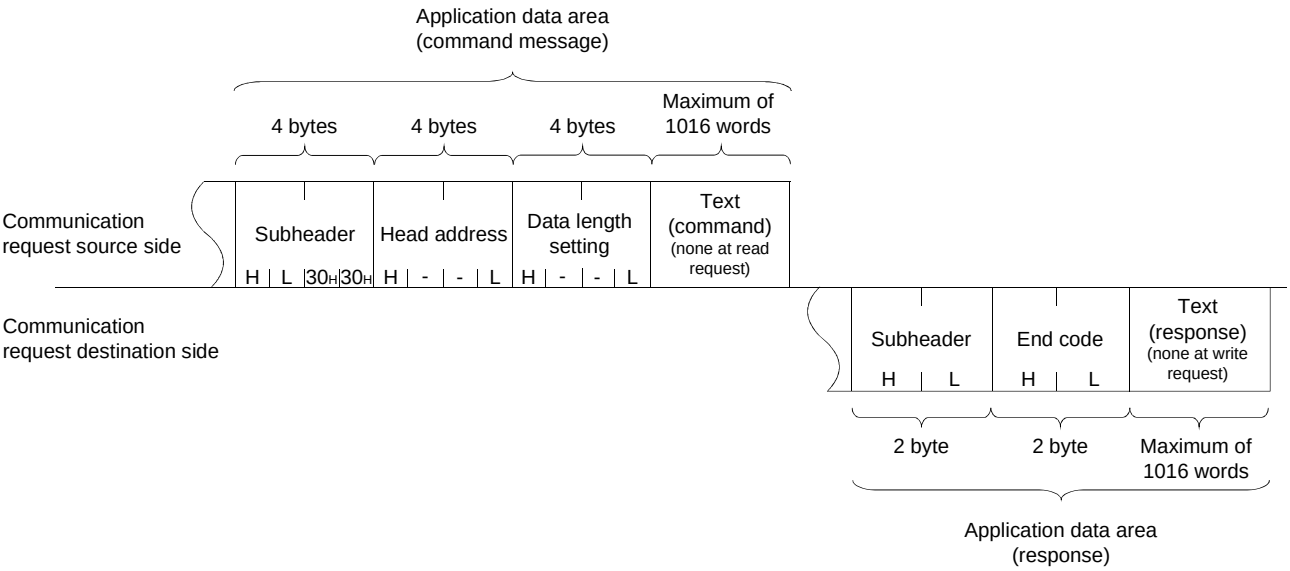
As shown below, the data code in the application data can be expressed in either binary or ASCII code. Switching between the binary code and the ASCII code is performed with GX Developer as follows.
[GX Developer] ― [Network parameters] ― [Operational settings] ― [Communication data code]
For more details, see Section 4.7, "Operational Settings".

(1) Format

(a) Communication using binary code



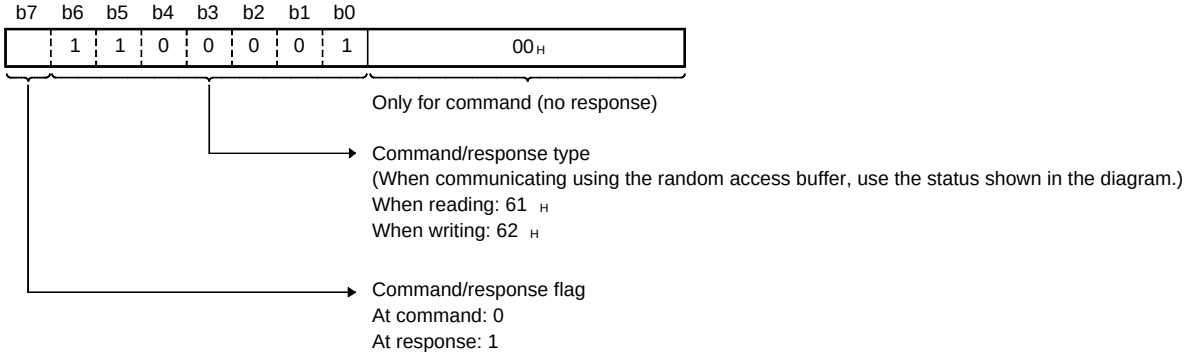
(b) Communication using ASCII code



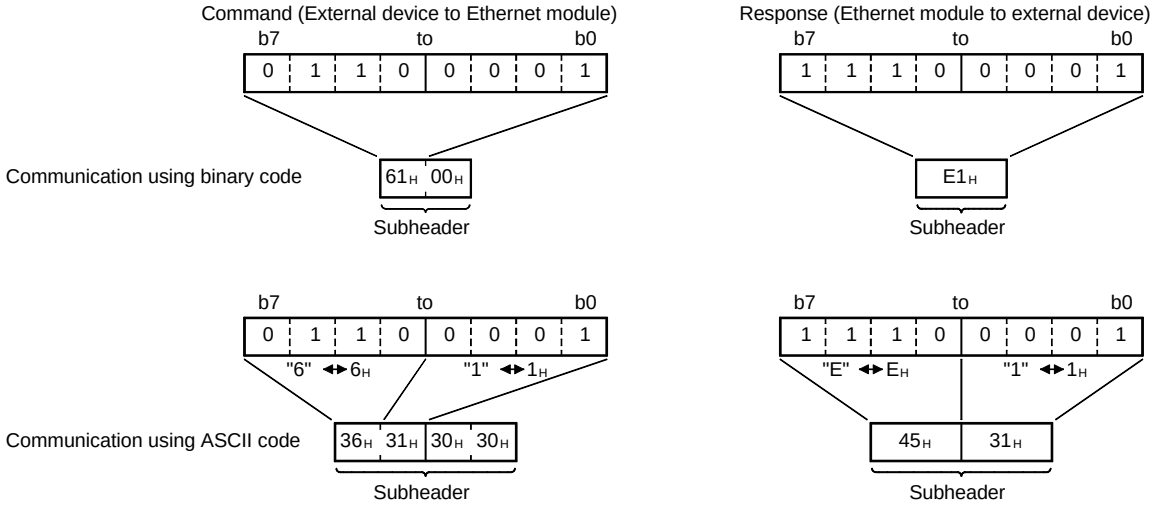
(2) Subheader

The format of the subheader is as shown below.

The user does not need to set the subheader when using the Ethernet module since the Ethernet module adds and deletes it automatically to the text.


b7 b6 b5 b4 b3 b2 b1 b0
1 1 0 0 0 0 1
00H
Only for command (no response)
Command/response type
(When communicating using the random access buffer, use the status shown in the diagram.)
When reading: 61 H
When writing: 62 H
Command/response flag
At command: 0
At response: 1

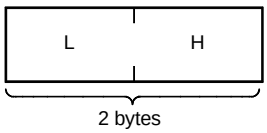
● When reading



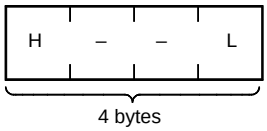
(3) Head address

Indicates the head address (ranging from 2680H to 3F7FH) of the random access buffer whose data is read/written from/to using its logical address (ranging from 0H to 17FFH … See Section 9.3).

(a) Communication using binary code : Designate the head address by its binary value.



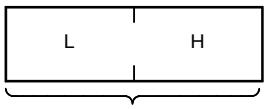
(b) Communication using ASCII code : Designate an ASCII code value that expresses the head address in hexadecimal.



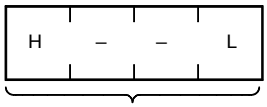
(4) Data length setting

Indicate the number of words of the read/written data in the random access buffer range.

(a) Communication using binary code : Designate the number of words by its binary value.



(b) Communication using ASCII code : Designate an ASCII code value that expresses the number of words in hexadecimal.



POINT

The data length can be designated in the following range:

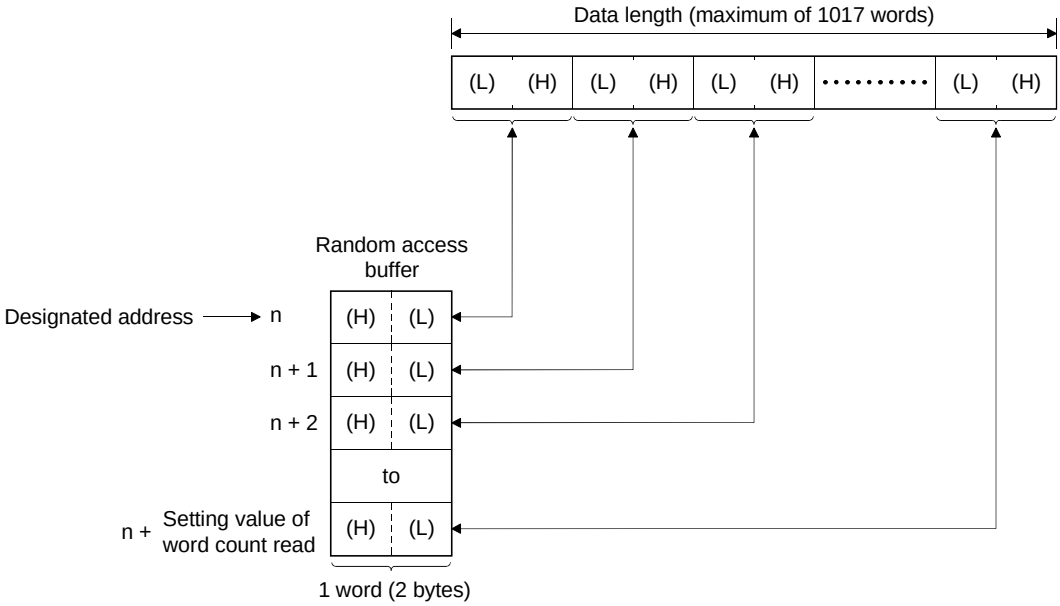
- Communication using binary code: Maximum of 1017 words
- Communication using ASCII code: Maximum of 508 words (*1)

*1 Since data is sent/received as ASCII data, the communication data size is approximately a half of the data size when using binary code.

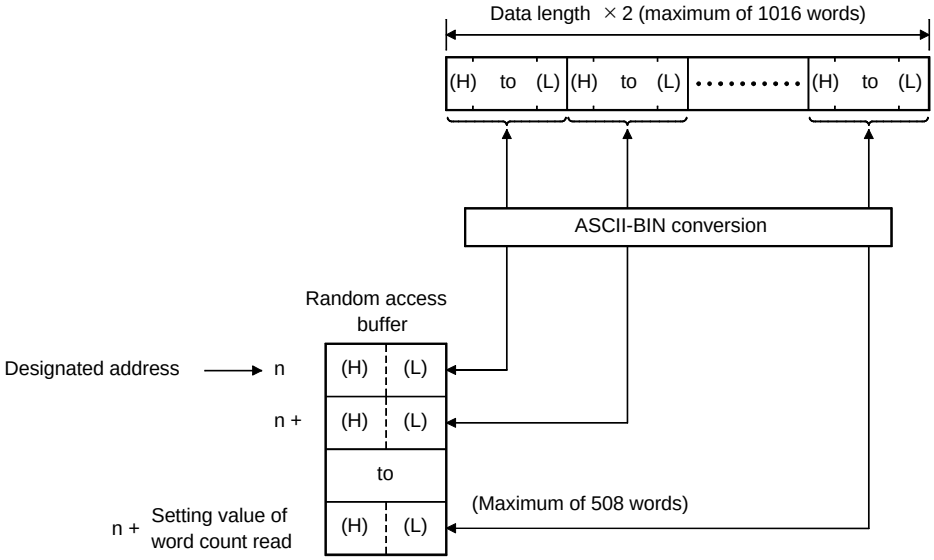
(5) Text

Indicate the data written to and read from the random access buffer.

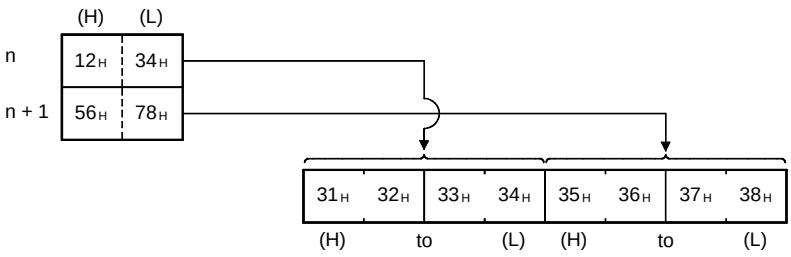
(a) Communication using binary code



(b) Communication using ASCII code



(Example)



9 - 9

(6) End codes

The following end codes are added to a response in random access buffer communication.

- Normal completion : 00H
- Abnormal completion : Value other than 00H (see Section 11.3.1)

End codes are stored in the communication status storage area of the buffer memory.

9 - 10

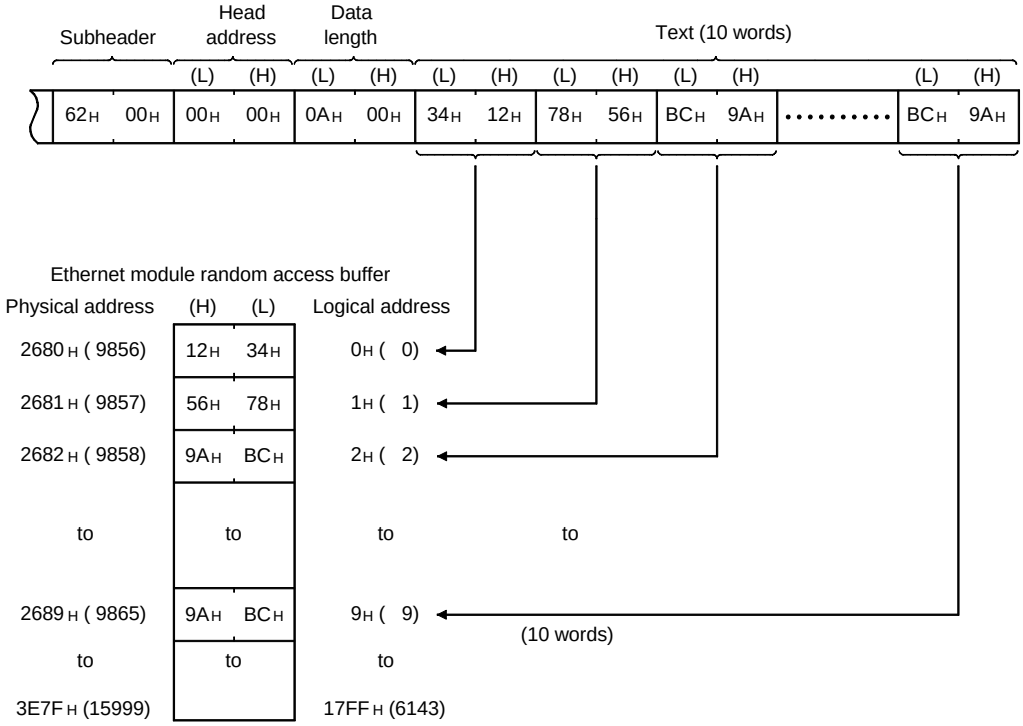
9.2.3 Examples of the command/response formats

This section explains examples of the command/response formats during communication using the random access buffer.

(1) Writing to the random access buffer by a write request from an external device

(a) Communication using binary code

1) Command format (external device to the Ethernet module)



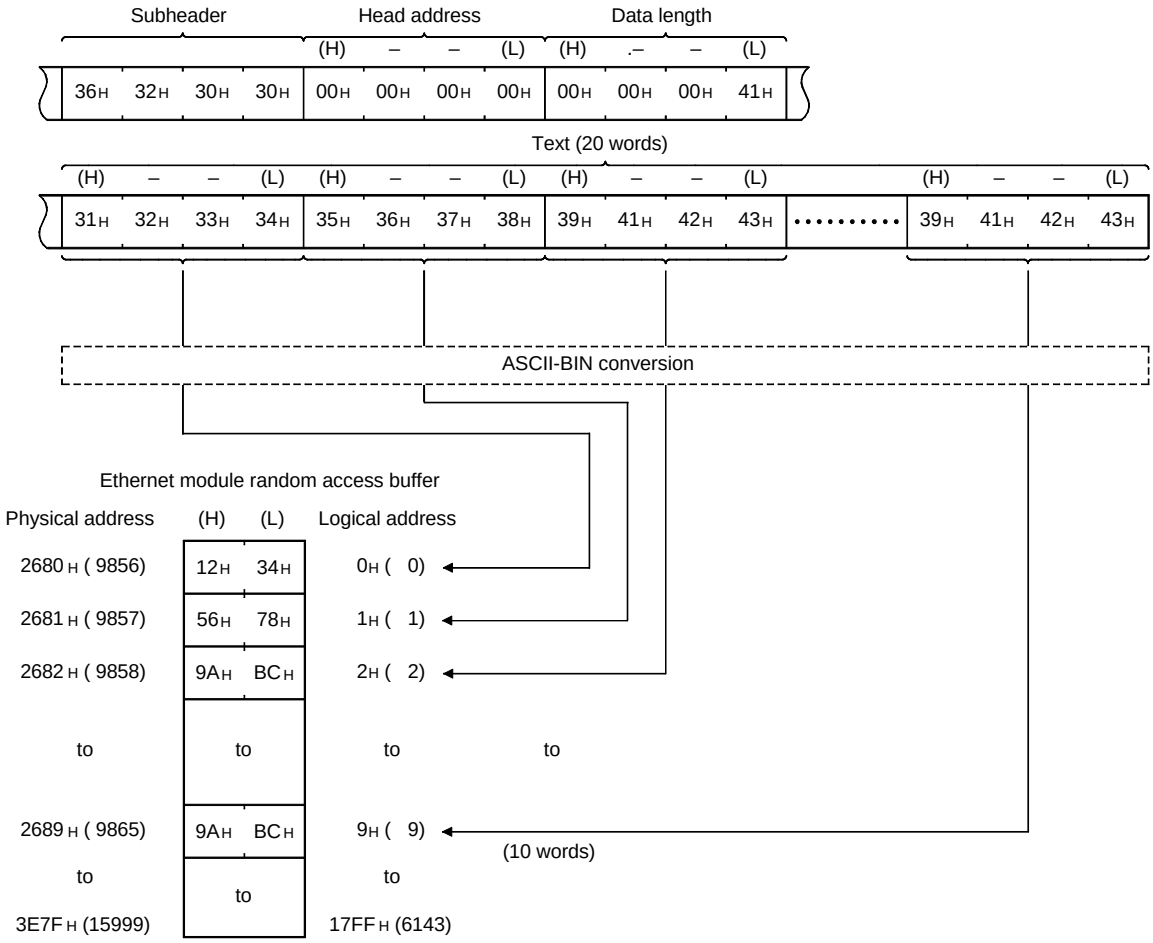
2) Response format (Ethernet module to the external device)

Subheader	End code
$E2_H$	00_H

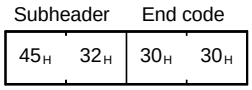
9 - 11

(b) Communication using ASCII code

1) Command format (external device to the Ethernet module)


Subheader
Head address
Data length
(H) – – (L) (H) .– – (L)
36H 32H 30H 30H 00H 00H 00H 00H 00H 00H 00H 41H
Text (20 words)
(H) – – (L) (H) – – (L) (H) – – (L) (H) – – (L)
31H 32H 33H 34H 35H 36H 37H 38H 39H 41H 42H 43H 39H 41H 42H 43H
ASCII-BIN conversion
Ethernet module random access buffer
Physical address (H) (L) Logical address
2680H (9856) 12H 34H 0H (0)
2681H (9857) 56H 78H 1H (1)
2682H (9858) 9AH BCH 2H (2)
to to to to
2689H (9865) 9AH BCH 9H (9)
(10 words)
to to to
3E7FH (15999) 17FFH (6143)

2) Response format (Ethernet module to the external device)

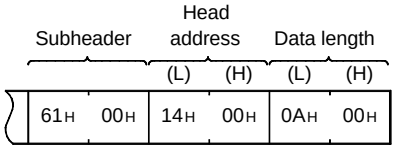


9 - 12

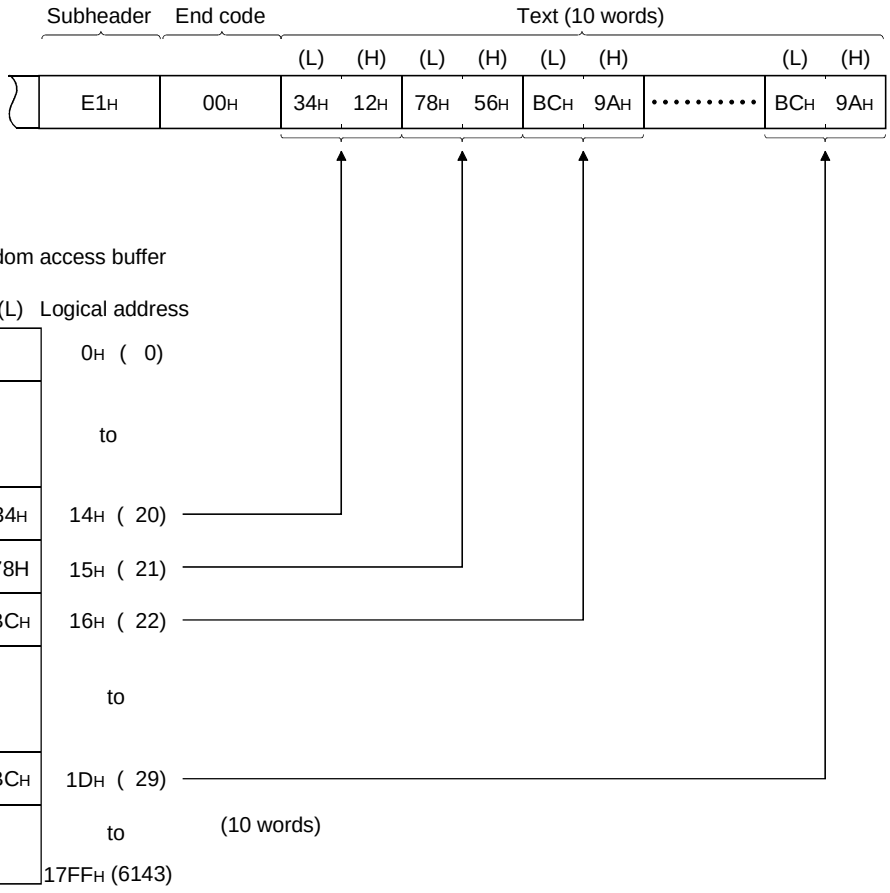
(2) Reading from the random access buffer by a read request from an external device

(a) Communication using binary code

1) Command format (external device to the Ethernet module)



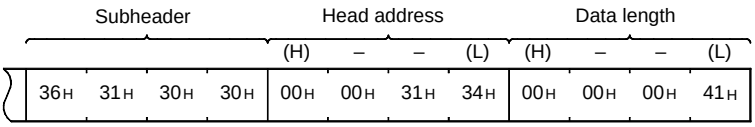
2) Response format (Ethernet module to the external device)


Subheader
End code
Text (10 words)
(L) (H) (L) (H) (L) (H) (L) (H)
E1H 00H 34H 12H 78H 56H BCH 9AH BCH 9AH
Ethernet module random access buffer
Physical address (H) (L) Logical address
2680H(9856) 0H (0)
to to to
2694H(9876) 12H 34H 14H (20)
2695H(9877) 56H 78H 15H (21)
2696H(9878) 9AH BCH 16H (22)
to to to
269DH (9885) 9AH BCH 1DH (29)
to to to (10 words)
3E7FH (15999) 17FFH (6143)

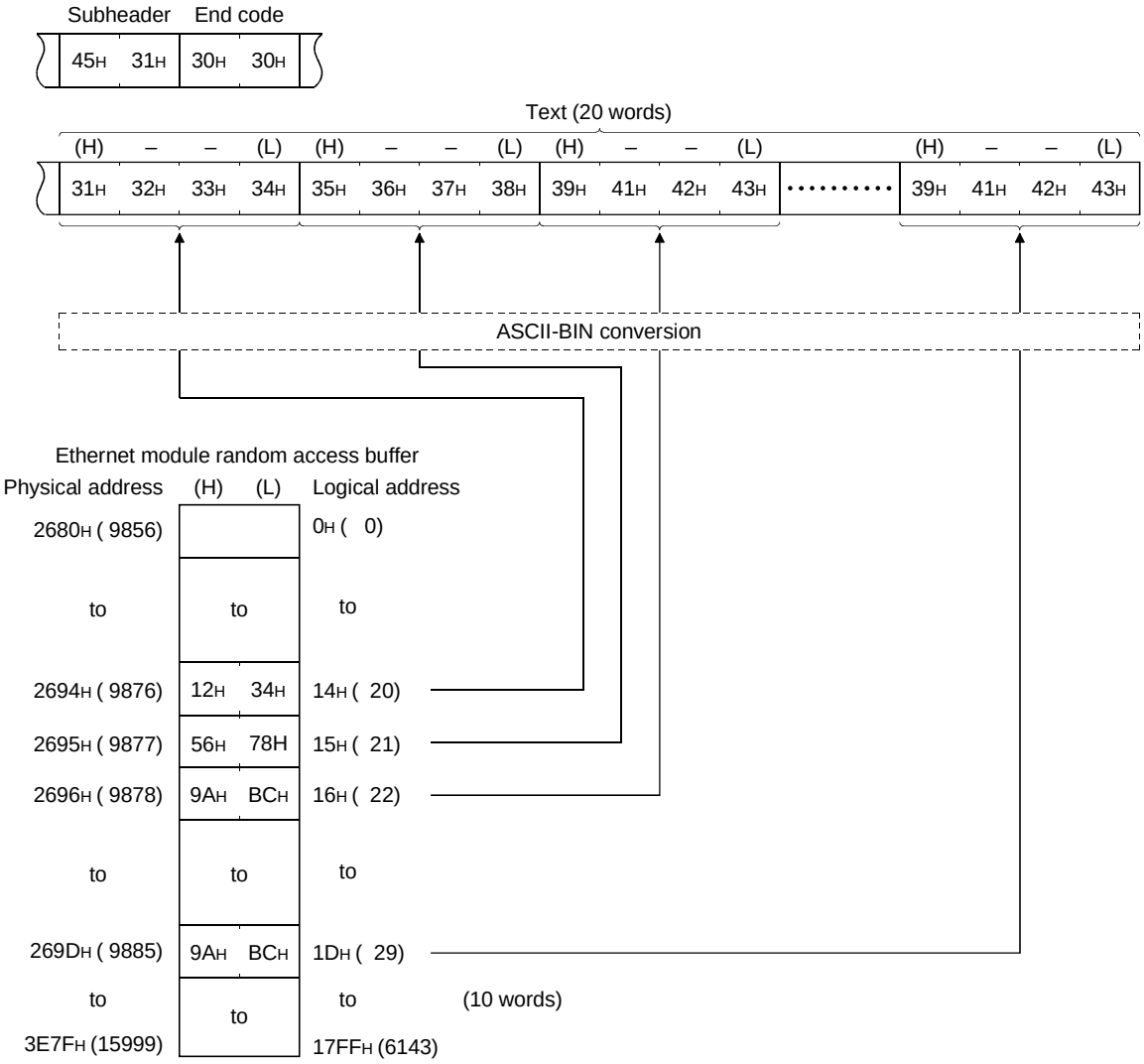
9 - 13

(a) Communication using ASCII code

1) Command format (external device to the Ethernet module)



2) Response format (Ethernet module to the external device)



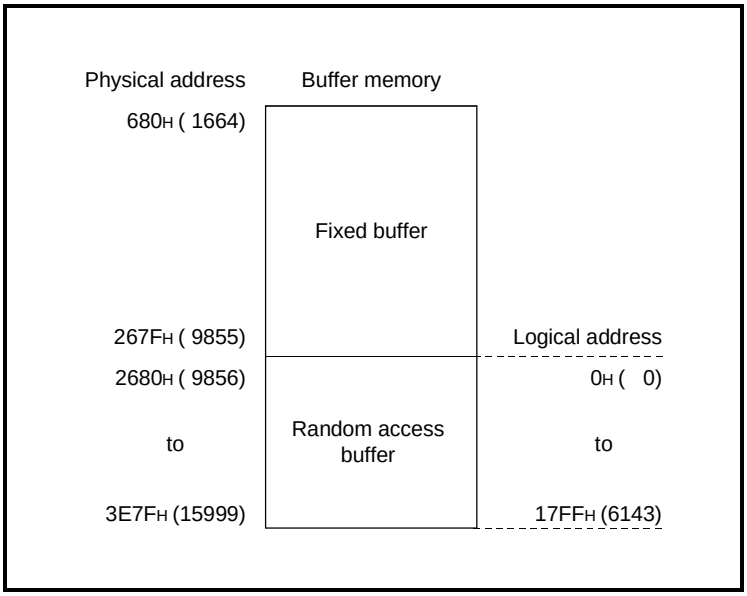
9 - 14

9.3 Physical and Logical Addresses of the Random Access Buffer

This section explains the head address of the Ethernet random access buffer (no battery backup), which is designated with the commands used for the random access buffer communication.

The designated addresses of the random access buffer are shown below.
Note that when designating the head address of the random access buffer, the address designated by an external device is different from that designated with the FROM/TO instructions in the sequence program.

Physical address : Address designated with the FROM/TO instructions of the sequence program
Logical address : Address designated by the external device as the head address item in the command



9.4 Precautions when Creating Programs

This section explains an outline of the precautions that should be observed when creating programs for data communications between the Ethernet module and external devices using the random access buffer.

(1) In order to communicate using the random access buffer, the initial processing and the connection open processing must be completed.

(2) The PLC CPU cannot issue a send request when communicating using the random access buffer.
Also, receive completion is not confirmed to the PLC CPU.
When it is necessary to synchronize data sending/receiving between the PLC CPU and an external device, use the fixed buffer communication function.

(3) The address designated for the random access buffer by an external device and the address designated with the FROM/TO instructions of the sequence program are different.
For more detail, see Section 9.3.

10 DEDICATED INSTRUCTIONS

Dedicated instructions are used to simplify programming for using the functions of the intelligent function module.
This chapter explains the dedicated instructions for the functions that are explained in this manual, among those dedicated commands available for the Ethernet module that can be used by the QCPU.

10.1 Dedicated Instruction List

The following table lists the dedicated instructions explained in this chapter:

Application	Dedicated instruction	Description of function	Reference section
For opening and closing connections	OPEN	Establishes a connection.	Section 10.8
	CLOSE	Disconnects a connection.	Section 10.5
For fixed buffer communication	BUFRCV	Reads received data. (For main program)	Section 10.2
	BUERCVS	Reads received data. (For interrupt program)	Section 10.3
	BURSND	Sends data.	Section 10.4
For reading and clearing error information	ERRCLR	Clears error information.	Section 10.6
	ERRRD	Reads error information.	Section 10.7
For reinitialization	UINI	Reinitializes the Ethernet module.	Section 10.9

POINT

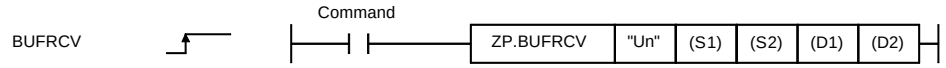
(1) The user should not change any data (control data, request data, etc.) that is specified with a dedicated instruction until the execution of that dedicated instruction is completed.
(2) All dedicated instructions must be executed online.
If any of the dedicated instructions is executed offline, no error will occur, but the execution of the dedicated instruction will not be completed.

10.2 BUFRCV Instruction

This instruction reads data received from an external device through fixed buffer communication.
This instruction is used in the main program.

Setting data	Applicable device								
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant	Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	–	○		–				○	–	–
(S2)	–	○		–				–	–	–
(D1)	–	○		–				–	–	–
(D2)	○	○		–				–	–	–

[Instruction code] [Executing condition]



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Connection number (1 to 16)		Binary 16 bits
(S2)	Head number of the devices that designate control data	System	Binary 16 bits
(D1)	Head number of the device that stores receive data		Binary 16 bits
(D2)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D2) + 1 also turns on if the instruction execution ends abnormally.		Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S2) + 0	System area	–	–	–
(S2) + 1	Complete status	• Stores the status at completion. 0000_H: Normal completion Other than 0000_H : Abnormal completion (error code) (*2)	–	System

Receive data

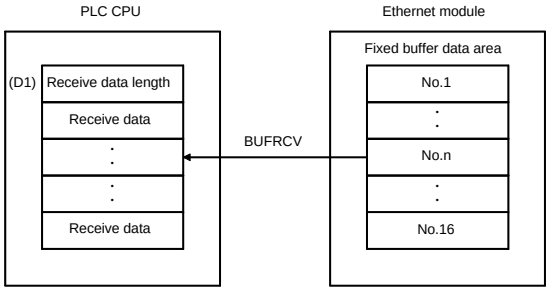
Device	Item	Setting data		Setting range	Set by (*1)
(D1) + 0	Receive data length	• Stores the data length of the data read from the fixed buffer data area in word units.		–	System
			There is a procedure (for communication using binary code): Number of words	1 to 1017	
			There is a procedure (for communication using ASCII code): Number of words	1 to 508	
			Non procedure (for communication using binary code): Number of bytes	1 to 2046	
(D1) + 1 to (D2) + n	Receive data	• Stores the data read from the fixed buffer data area sequentially in ascending order.		–	System

*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

Functions

(1) This instruction reads receive data (for fixed buffer communication) from the connection specified by (S1) for the module designated by Un.

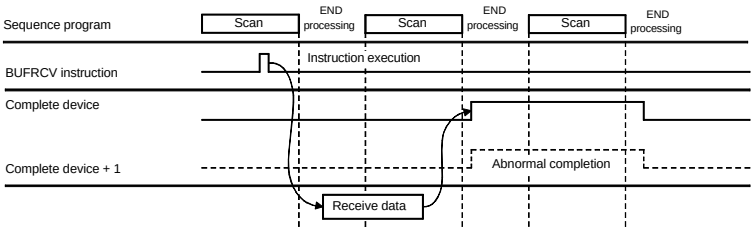


(2) Whether or not the BUFRCV instruction has been completed can be checked by the complete bit devices (D2) + 0 and (D2) + 1.

(a) Complete bit device (D2) + 0
Turns on at the END processing of the scan where the BUFRCV instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D2) + 1
Turns on and off depending on the completion status of the BUFRCV instruction.
- Normal completion : Stays off and does not change.
- Abnormal completion : Turns on at the END processing of the scan where the BUFRCV instruction is completed, and turns off at the next END processing.

[Operation when the BUFRCV instruction is being executed]



(3) The ZP.BUFRCV instruction is executed when the read instruction (indicated by a bit for the applicable connection in the fixed buffer receive status signal storage area (address: 5005_H) of the buffer memory) switches from off to on.

(4) When reading receive data from the same connection, this cannot be used together with BUFRCVS instructions (for interrupt programs).

Errors

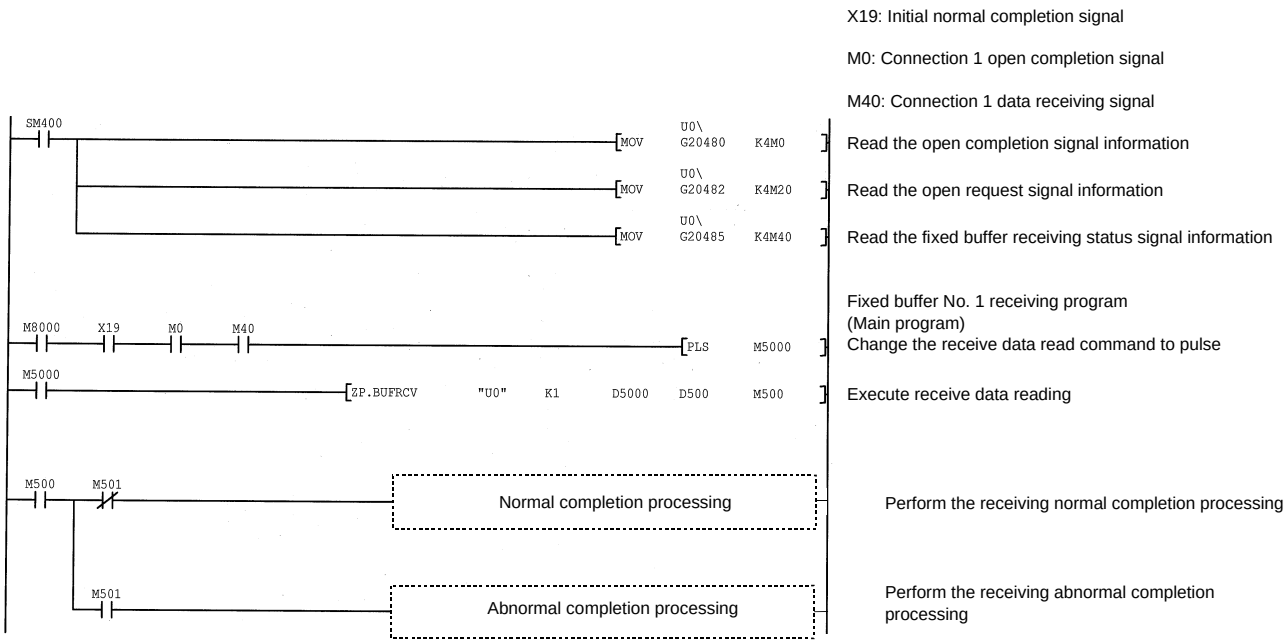
(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D2) + 1, turns on and the error code is stored in the complete status area (S2) + 1. Refer to the following manuals regarding the error codes, check the errors and take corrective actions.

<Error codes>

$4FFF_H$ or less	: QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)
$C000_H$ or higher	: Section 11.3.3 of this manual

Program example

A program that reads receive data from the fixed buffer for connection number 1:
When the input/output signals of the Ethernet module are X/Y00 to X/Y1F

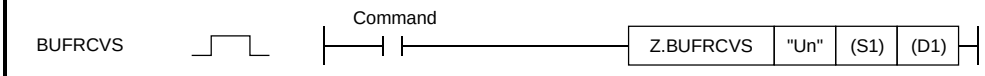

X19: Initial normal completion signal
M0: Connection 1 open completion signal
M40: Connection 1 data receiving signal
SM400
MOV U0\G20480 K4M0
Read the open completion signal information
MOV U0\G20482 K4M20
Read the open request signal information
MOV U0\G20485 K4M40
Read the fixed buffer receiving status signal information
Fixed buffer No. 1 receiving program
(Main program)
M8000 X19 M0 M40
PLS M5000
Change the receive data read command to pulse
M5000
ZP.BUFRCV "U0" K1 D5000 D500 M500
Execute receive data reading
M500 M501
Normal completion processing
Perform the receiving normal completion processing
M501
Abnormal completion processing
Perform the receiving abnormal completion processing

10.3 BUFRCVS Instruction

This instruction reads data received from an external device through fixed buffer communication.
This instruction is used in the interrupt program.

Setting data	Applicable device									
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	–	○		–				○	–	–
(D1)	–	○		–				–	–	–

[Instruction code] [Executing condition]



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Connection number (1 to 16)		Binary 16 bits
(D1)	Head number of the device that stores receive data	System	Binary 16 bits

The file registers of each of the local device and the program cannot be used as device to be used in the setting data.

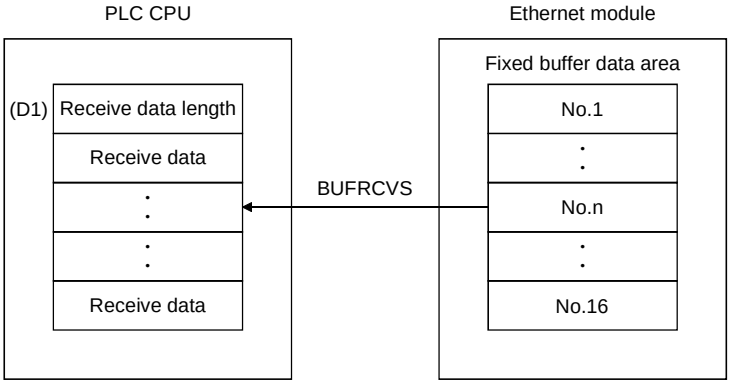
Receive data

Device	Item	Setting data		Setting range	Set by (*1)
(D1) + 0	Receive data length	• Stores the data length of the data read from the fixed buffer data area in word units.		–	System
			There is a procedure (for communication using binary code): Number of words	1 to 1017	
			There is a procedure (for communication using ASCII code): Number of words	1 to 508	
			Non procedure (for communication using binary code): Number of bytes	1 to 2046	
(D1) + 1 to (D1) + n	Receive data	• Stores the data read from the fixed buffer data area sequentially in ascending order.		–	System

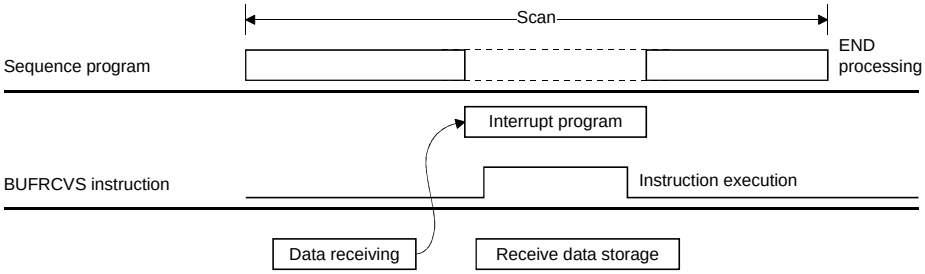
*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

Functions

(1) This instruction reads receive data (for fixed buffer communication) from the connection specified by (S1) for the module designated by Un.



[Operation when the BUFRCVS instruction is being executed]



(2) The Z.BUFRCVS instruction is used by the interrupt programs and its processing completes within one scan.

(3) In order to read receive data with an interrupt program, it is necessary to perform both the interrupt settings and interrupt pointer settings with parameter settings of GX Developer.

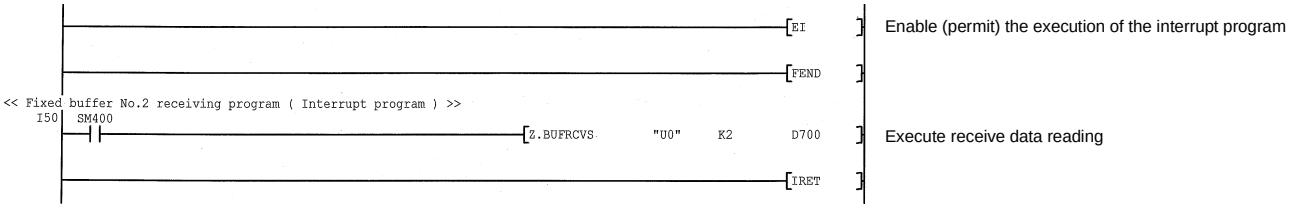
(4) When reading receive data from the same connection, this cannot be used together with BUFRCV instructions (for main programs).

Errors

(1) When the dedicated instruction is completed abnormally, the error flag (SM0) turns on and the error code is stored in SD0.
Refer to the following manuals regarding the error code, check the errors and take corrective actions.
<Error codes>
4FFF$_H$ or less : QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)
C000$_H$ or higher : Section 11.3.3 of this manual

Program example

A program that reads receive data from the fixed buffer for connection number 2:
When the input/output signals of the Ethernet module are X/Y00 to X/Y1F



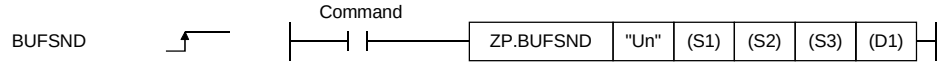
10 - 7

10.4 BUFSND Instruction

This instruction send data to an external device through fixed buffer communication.

Setting data	Applicable device								
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	—	○		—				○	—	—
(S2)	—	○		—				—	—	—
(S3)	—	○		—				—	—	—
(D1)	○	○		—				—	—	—

[Instruction code] [Executing condition]



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Connection number (1 to 16)		Binary 16 bits
(S2)	Head number of the device that stores control data	System	Binary 16 bits
(S3)	Head number of the device that stores send data	User	Binary 16 bits
(D1)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D1) + 1 also turns on if the instruction execution ends abnormally.	System	Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S2) + 0	System area	—	—	—
(S2) + 1	Complete status	• Stores the status at completion. 0000_H: Normal completion Other than 0000_H : Abnormal completion (error code) (*2)	—	System

Send data

Device	Item	Setting data		Setting range	Set by (*1)
(S3) + 0	Send data length	• Designate the send data length in word units.		—	User
			There is a procedure (for communication using binary code): Number of words	1 to 1017	
			There is a procedure (for communication using ASCII code): Number of words	1 to 508	
			Non procedure (for communication using binary code): Number of bytes	1 to 2046	
(S3) + 1 to (S3) + n	Send data	• Designate the send data.		—	User

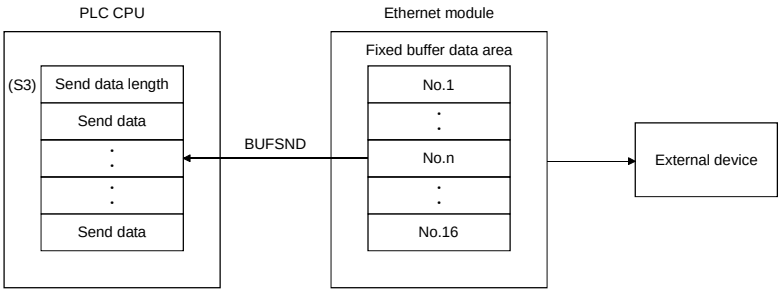
*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

10 - 8

Functions

(1) This instruction sends the data designated by (S3) to the external device of the connection specified by (S1) for the module designated by Un.

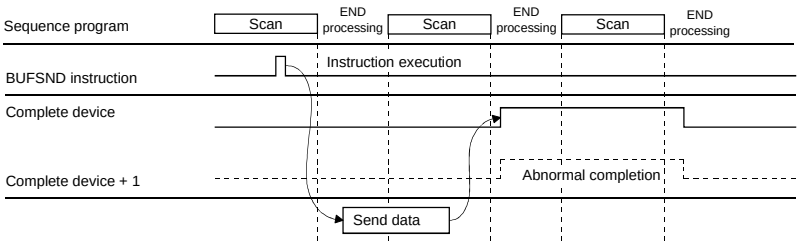


(2) Whether or not the BUFSND instruction has been completed can be checked by the complete bit devices (D1) + 0 and (D1) + 1.

(a) Complete bit device (D1) + 0
Turns on at the END processing of the scan where the BUFSND instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D1) + 1
Turns on and off depending on the completion status of the BUFSND instruction.
- Normal completion : Stays off and does not change.
- Abnormal completion : Turns on at the END processing of the scan where the BUFSND instruction is completed, and turns off at the next END processing.

[Operation when the BUFSND instruction is being executed]



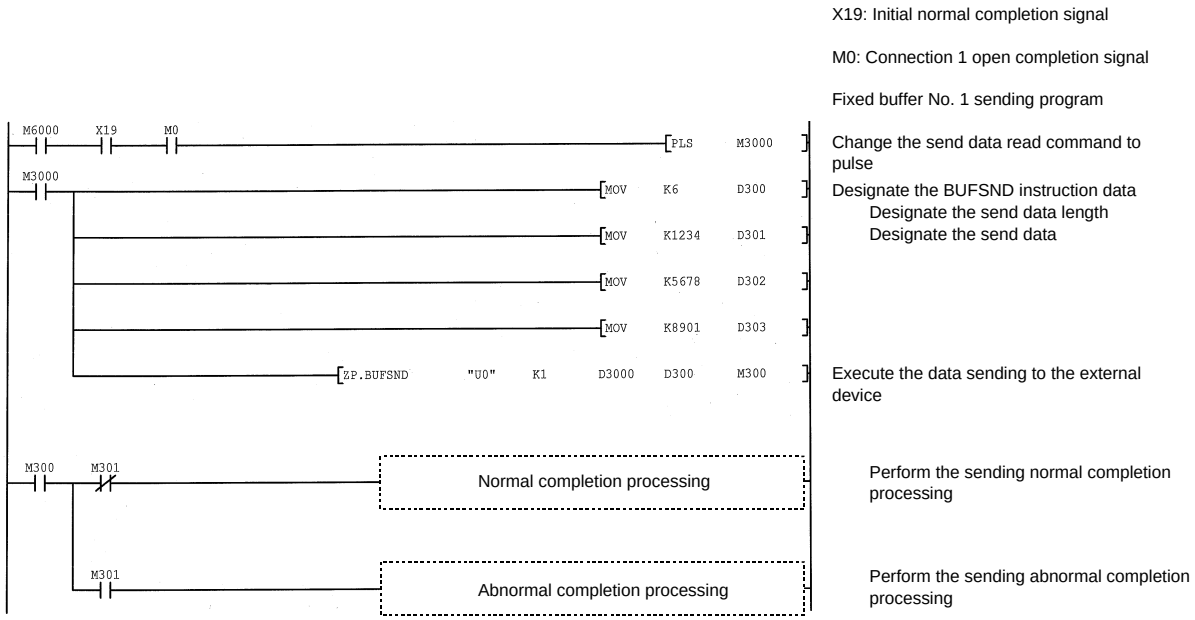
(3) The ZP.BUFSND instruction is executed when the send instruction switches from off to on.

Errors

(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D1) + 1, turns on and the error code is stored in the complete status area (S2) + 1. Refer to the following manuals regarding the error code, check the errors and take corrective actions.
<Error codes>
4FFF$_H$ or less : QCPU (Q mode) User's Manual (Hardware Design, Maintenance and Inspection)
C000$_H$ or higher : Section 11.3.3 of this manual

Program example

A program that sends data from the fixed buffer of connection number 1:
When the input/output signals of the Ethernet module are X/Y00 to X/Y1F

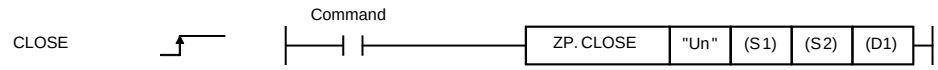


10.5 CLOSE Instruction

This instruction disconnects (closes) a connection by which data was communicated with an external device.

Setting data	Applicable device									
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	–	○		–				○	–	–
(S2)	–	○		–				–	–	–
(D1)	○	○		–				–	–	–

[Instruction code] [Executing condition]



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Connection number (1 to 16)		Binary 16 bits
(S2)	Head number of the device that stores control data	System	Binary 16 bits
(D1)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D1) + 1 also turns on if the instruction execution ends abnormally.		Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S2) + 0	System area	–	–	–
(S2) + 1	Complete status	• Stores the status at completion. 0000_H: Normal completion Other than 0000_H : Abnormal completion (error code) (*2)	–	System

*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

10 - 11

Functions

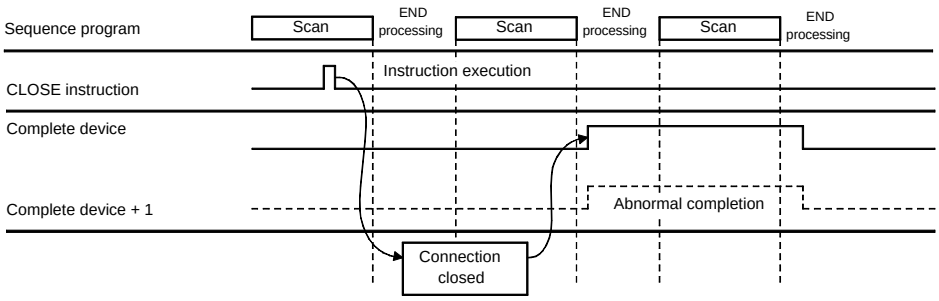
(1) This instruction closes the connection specified by (S1) for the module designated by Un (disconnecting connections).

(2) Whether or not the CLOSE instruction has been completed can be checked by the complete bit devices (D1) + 0 and (D1) + 1.

(a) Complete bit device (D1) + 0
Turns on at the END processing of the scan where the CLOSE instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D1) + 1
Turns on and off depending on the completion status of the CLOSE instruction.
- Normal completion : Stays off and does not change.
- Abnormal completion : Turns on at the END processing of the scan where the CLOSE instruction is completed, and turns off at the next END processing.

[Operation when the CLOSE instruction is being executed]



(3) The ZP.CLOSE is executed when the close instruction switches from off to on.

Important

Never execute the open/close processing using input/output signals and the open/close processing using the OPEN or CLOSE dedicated instructions simultaneously for the same connection. It will result in malfunctions.

Errors

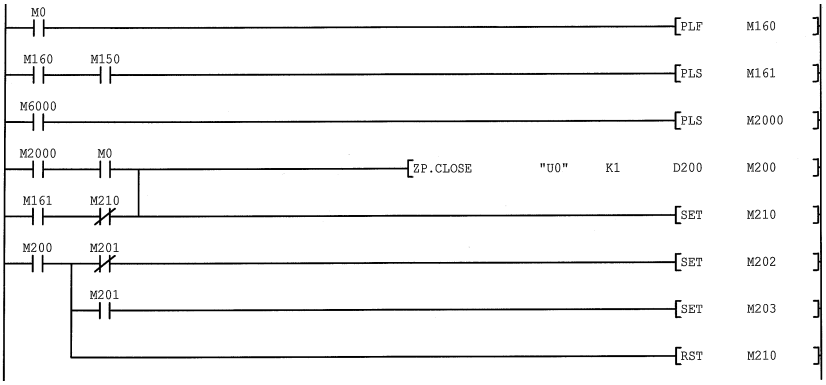
(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D1) + 1, turns on and the error code is stored in the complete status area (S2) + 1. Refer to the following manuals regarding the error codes, check the errors and take corrective actions.
<Error codes>
4FFF$_H$ or less : QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)
C000$_H$ or higher : Section 11.3.3 of this manual

Program example

A program that closes the connection number 1:
When the input/output signals of the Ethernet module are X/Y00 to X/Y1F

M0: Connection 1 open completion signal

M150: OPEN instruction execution normal completion flag

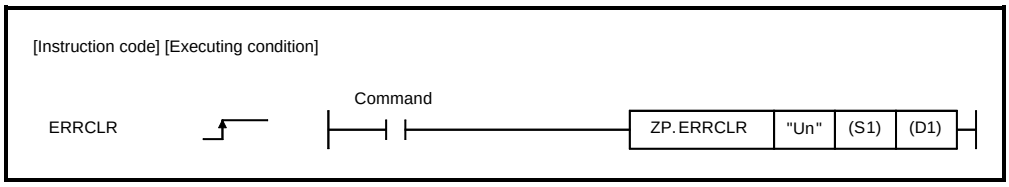


10 - 13

10.6 ERRCLR Instruction

This instruction turns off the LEDs of the Ethernet module and clears data information stored in the buffer memory.

Setting data	Applicable device									
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	—	○		—				—	—	—
(D1)	○	○		—				—	—	—



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Head number of the device that stores control data	User, system	Binary 16 bits
(D1)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D1) + 1 also turns on if the instruction execution ends abnormally.	System	Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S1) + 0	System area	—	—	—
(S1) + 1	Complete status	• Stores the status at completion. 0000_H: Normal completion Other than 0000_H : Abnormal completion (error code) (*2)	—	System
(S1) + 2	Clearing target designation	• Designate the error information to be cleared. 0000_H: Initial abnormal code 0001_H to 0016_H : Open abnormal code for the applicable connection 0100_H: Error log block area 0101_H: Communication status – status by each protocol 0102_H: Communication status – E-mail receive status 0103_H: Communication status – E-mail send status $FFFF_H$: Clears all of the above	(as described in the left)	User
(S1) + 3	Clearing function designation	• Designate the function to be cleared. 0000_H: [COM.ERR] LED off, error code clear $FFFF_H$: Error log clear	0000_H $FFFF_H$	User
(S1) + 4 to (S1) + 7	System area	—	—	—

*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

10 - 14

Functions

(1) This instruction turns [COM.ERR] LEDs off and clears error information listed below for the module designated by Un.

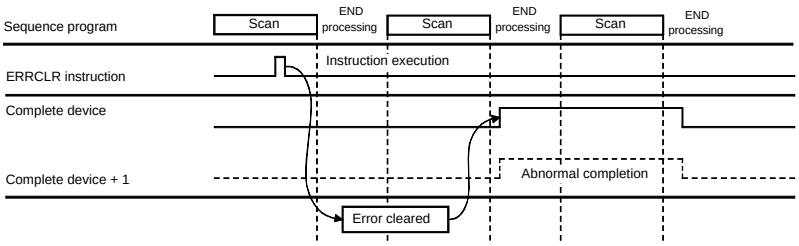
Target item		Target designation (S1) + 2	Function designation (S1) + 3	Error information to be cleared (buffer memory)
Initial error		0000_H	0000_H	• Initial abnormal code (address: 69_H) • [COM.ERR] LED off
Open error		0001_H to 0016_H	0000_H	• Open abnormal code for applicable connection (address: $7C_H$, 86_H…) • [COM.ERR] LED off
Error log		0100_H	$FFFF_H$	• Error log (address:$E3_H$ to 174_H)
Communication status	Status by each protocol	0101_H	$FFFF_H$	• Clears communication status (address: 178_H to $1FF_H$)
	E-mail receive status	0102_H	$FFFF_H$	• E-mail receiving (address: 5871_H to $5B38_H$)
	E-mail send status	0103_H	$FFFF_H$	• E-mail sending (address: $5B39_H$ to $5CA0_H$)
All		$FFFF_H$	$FFFF_H$	• Clears all of the above.

(2) Whether or not the ERRCLR instruction has been completed can be checked by the complete bit devices (D1) + 0 and (D1) + 1.

(a) Complete bit device (D1) + 0
Turns on at the END processing of the scan where the ERRCLR instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D1) + 1
Turns on and off depending on the completion status of the ERRCLR instruction.
- Normal completion : Stays off and does not change.
- Abnormal completion: Turns on at the END processing of the scan where the ERRCLR instruction is completed, and turns off at the next END processing.

[Operation when the ERRCLR instruction is being executed]



(3) The ZP.ERRCLR function is executed when the clear instruction switches from off to on.

10 - 15

Errors

(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D1) + 1, turns on and the error code is stored in the complete status area (S1) + 1. Refer to the following manuals regarding the error code, check the errors and take corrective actions.

<Error codes>

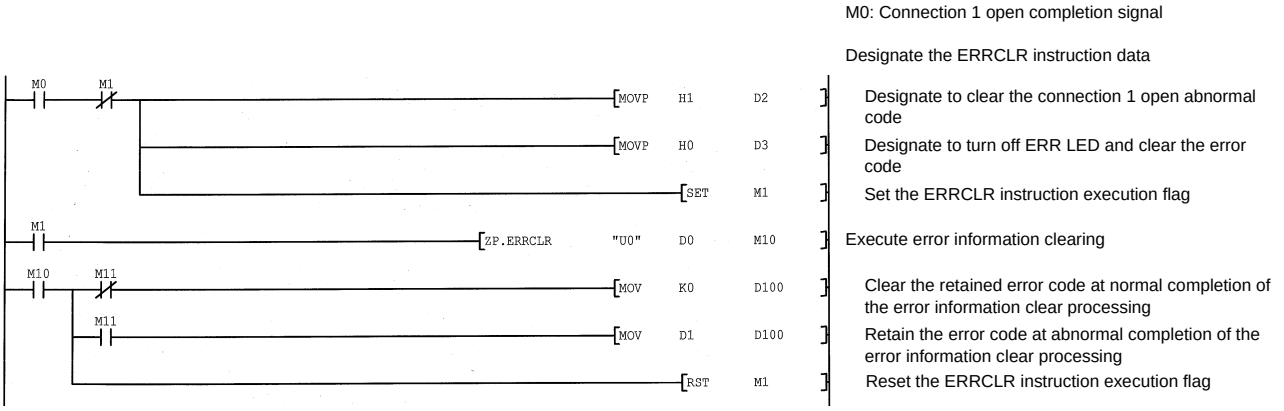
4FFF$_H$ or less : QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)

C000$_H$ or higher : Section 11.3.3 of this manual

Program example

A program that clears the open abnormal code for connection 1:

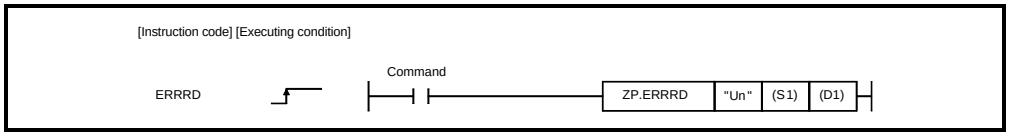
When the input/output signals of the Ethernet module are X/Y00 to X/Y1F


M0: Connection 1 open completion signal
Designate the ERRCLR instruction data
M0
M1
MOVP H1 D2
Designate to clear the connection 1 open abnormal code
MOVP H0 D3
Designate to turn off ERR LED and clear the error code
SET M1
Set the ERRCLR instruction execution flag
M1
ZP.ERRCLR "U0" D0 M10
Execute error information clearing
M10
M11
MOV K0 D100
Clear the retained error code at normal completion of the error information clear processing
M11
MOV D1 D100
Retain the error code at abnormal completion of the error information clear processing
RST M1
Reset the ERRCLR instruction execution flag

10.7 ERRRD Instruction

This instruction reads the error information stored in the buffer memory of the Ethernet module.

Setting data	Applicable device								
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant	Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	—	○		—				—	—	—
(D1)	○	○		—				—	—	—



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Head number of the device that stores control data	User, system	Binary 16 bits
(D1)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D1) + 1 also turns on if the instruction execution ends abnormally.	System	Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S1) + 0	System area	—	—	—
(S1) + 1	Complete status	• Stores the status at completion. 0000_H: Normal completion Other than 0000_H : Abnormal completion (error code) (*2)	—	System
(S1) + 2	Read information designation	• Designates the error information to read. 0 : Initial abnormal code 1 to 16: Open abnormal code for the applicable connection	0 1 to 16	User
(S1) + 3	Read target information designation	• Designates a target of the error information to read. 0000_H: Information of the last error that has occurred	0000_H	User
(S1) + 4	Error information	• Stores the error information that has been read. 000_H : No error Other than 0000_H: Error code (*2)	0000_H	System
(S1) + 5 to (S1) + 7	System area	—	—	—

*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

Functions

(1) This instruction reads the error code and information in the error log of the module designated by Un.

Target item	Target designation (S1) + 2	Function designation (S1) + 3	Error information to be cleared (buffer memory)
Initial error	0000_H	0_H	• Initial abnormal code (address: 69_H)
Open error	0001_H to 0016_H	0_H	• Open abnormal code for applicable connection (address: $7C_H$, 86_H…)

10 - 17

(2) Whether or not the ERRRD instruction has been completed can be checked by the complete bit devices (D1) + 0 and (D1) + 1.

(a) Complete bit device (D1) + 0

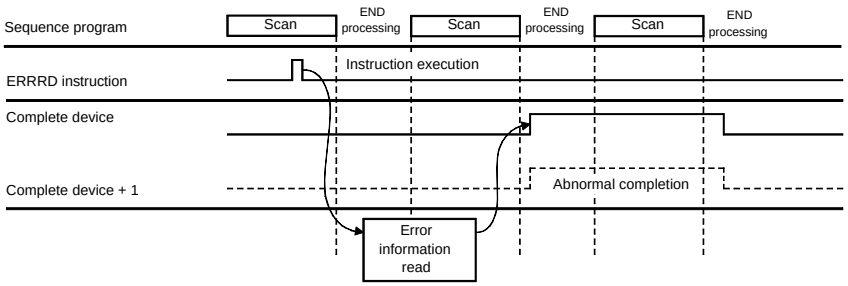
Turns on at the END processing of the scan where the ERRRD instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D1) + 1

Turns on and off depending on the completion status of the ERRRD instruction.

- Normal completion : Stays off and does not change.
- Abnormal completion : Turns on at the END processing of the scan where the ERRRD instruction is completed, and turns off at the next END processing.

[Operation when the ERRRD instruction is being executed]



(3) The ZP.ERRRD is executed when the read instruction switches from off to on.

Errors

(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D1) + 1, turns on and the error code is stored in the complete status area (S1) + 1. Refer to the following manuals regarding the error code, check the errors and take corrective actions.

<Error codes>

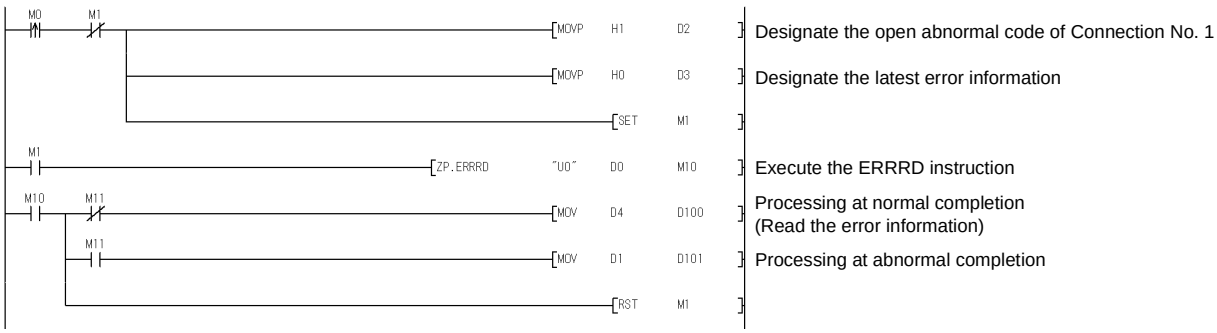
4FFF$_H$ or less : QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)

C000$_H$ or higher : Section 11.3.3 of this manual

Program example

A program to read the open abnormal code for connection 1:

When the input/output signals of the Ethernet module are X/Y00 to X/Y1F

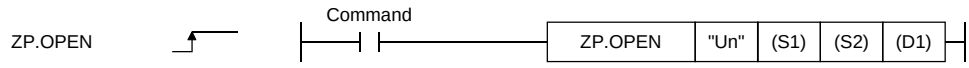


10.8 OPEN Instruction

This instruction establishes a connection (open processing) with an external device to perform data communication.

Setting data	Applicable device								
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	—	○		—				○	—	—
(S2)	—	○		—				—	—	—
(D1)	○	○		—				—	—	—

[Instruction code] [Executing condition]



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Connection number (1 to 16)		Binary 16 bits
(S2)	Head number of the device that stores control data	User, system	Binary 16 bits
(D1)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D1) + 1 also turns on if the instruction execution ends abnormally.	System	Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

10 - 19

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S2) + 0	Execution type/complete type	• Designate which settings to use at open processing of a connection, either the parameter setting values from GX Developer or the setting values of the control data starting from (S2) + 2. 0000_H: Open processing with parameters set in [Open settings] of GX Developer 8000_H: Open processing with parameters designated with control data from (S2) + 2 to (S2) + 9.	0000_H 8000_H	User
(S2) + 1	Complete status	• Stores the status at completion. 0000_H: Normal completion Other than 0000_H : Abnormal completion (error code) (*2)	—	System
(S2) + 2	Application setting area	• Designate how to use a connection. b15 b14 b13 to b10 b9 b8 b7 b6 to b2 b1 b0 6) 0 5) 4) 3) 0 2) 1) 1) Usage of fixed buffer 0: Sending or fixed buffer communication is not executed 1: For receiving 2) Destination existence confirmation 0: No confirm 1: Confirm 3) Pairing open setting 0: No pairs 1: Pairs 4) Communication method (protocol) 0: TCP/IP 1: UDP/IP 5) Fixed buffer communication 0: Procedure exist 1: No procedure 6) Open system 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open	(as described in the left)	User
(S2) + 3	local station Port No.	• Designate the port No. of the local station	407_H to 1387_H $138B_H$ to $FFFE_H$	User
(S2) + 4 (S2) + 5	Destination IP address	• Designate the IP address of the external device.	1_H to $FFFFFFFF_H$ ($FFFFFFFF_H$: simultaneous broadcast)	User
(S2) + 6	Destination Port No.	• Designate the port No. of the external device.	401_H to $FFFF_H$ ($FFFF_H$: simultaneous broadcast)	User
(S2) + 7 to (S2) + 9	Destination Ethernet address	• Designate the Ethernet address of the external device.(See Section 5.5 POINT)	n 000000000000_H $FFFFFFFFFFFF_H$	User

*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

10 - 20

Functions

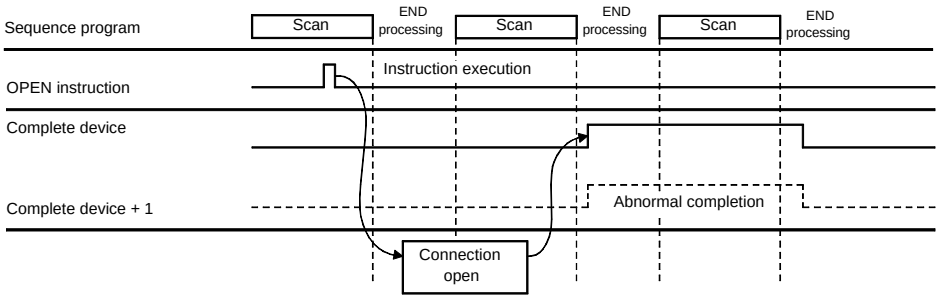
(1) This instruction performs the open processing for a connection specified by (S1) for the module designated by Un.
The selection of the setting values to be used for the open processing is designated by (S2) + 0.

(2) Whether or not the OPEN instruction has been completed can be checked by the complete bit devices (D1) + 0 and (D1) + 1.

(a) Complete bit device (D1) + 0
Turns on at the END processing of the scan where the OPEN instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D1) + 1
Turns on and off depending on the completion status of the OPEN instruction.
- Normal completion : Stays off and does not change.
- Abnormal completion : Turns on at the END processing of the scan where the OPEN instruction is completed, and turns off at the next END processing.

[Operation when the OPEN instruction is being executed]



(3) The ZP.OPEN is executed when the open instruction switches from off to on.

POINT
Never execute the open/close processing using input/output signals and the open/close processing using the OPEN or CLOSE dedicated instruction simultaneously for the same connection. It will result in malfunctions.

10 - 21

Errors

(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D1) + 1, turns on and the error code is stored in the complete status area (S2) + 1. Refer to the following manuals regarding the error code, check the errors and take corrective actions.

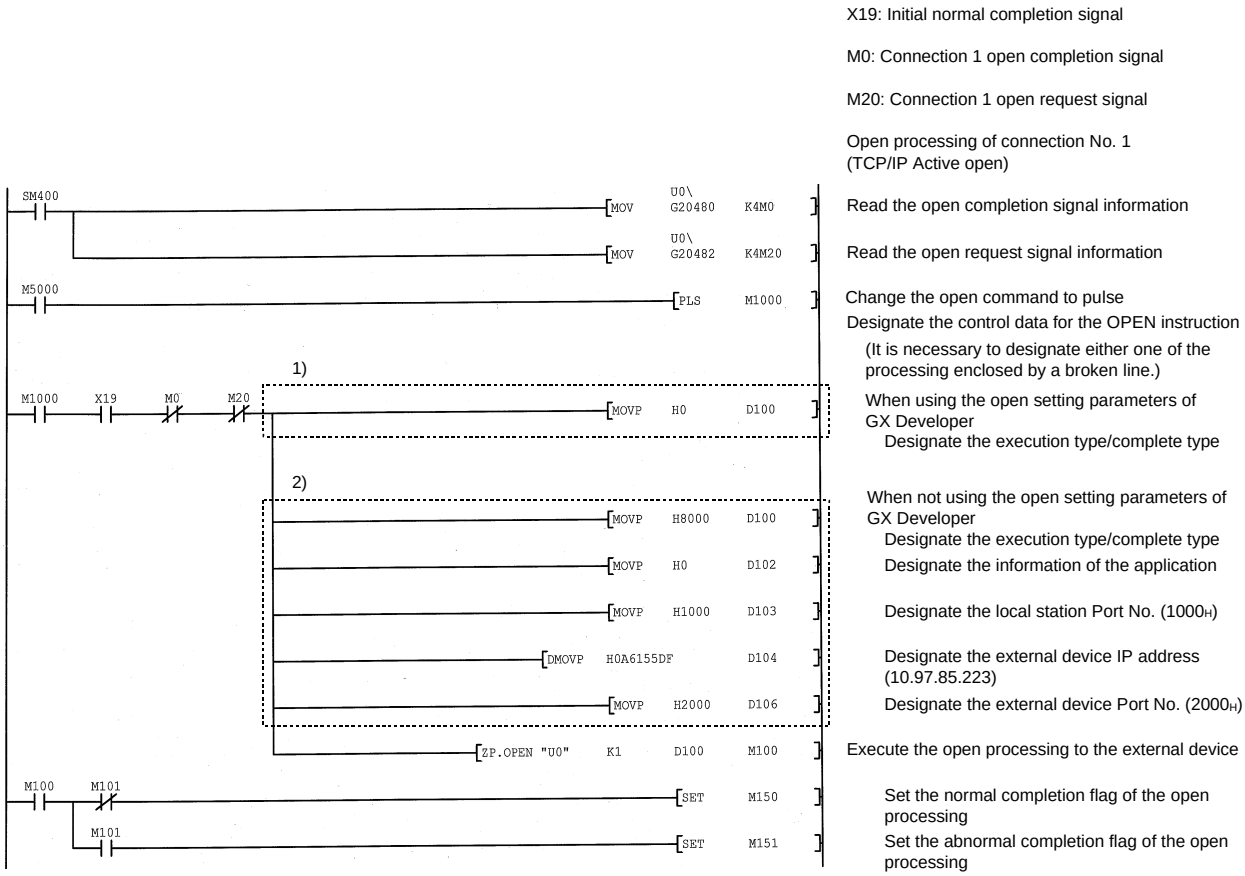
<Error codes>

4FFF$_H$ or less : QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)

C000$_H$ or higher : Section 11.3.3 of this manual

Program example

A program that Active open the connection 1 for TCP/IP communication:
When the input/output signals of the Ethernet module are X/Y00 to X/Y1F

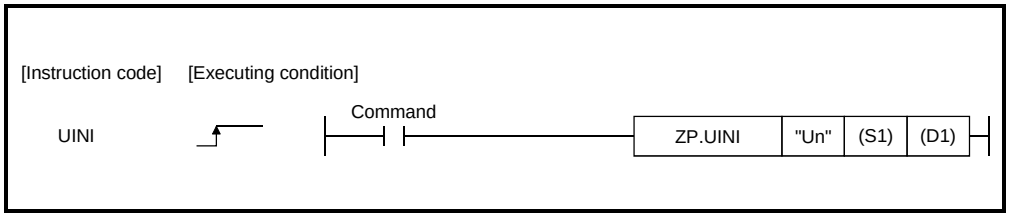

X19: Initial normal completion signal
M0: Connection 1 open completion signal
M20: Connection 1 open request signal
Open processing of connection No. 1 (TCP/IP Active open)
SM400
MOV U0\G20480 K4M0
Read the open completion signal information
MOV U0\G20482 K4M20
Read the open request signal information
M5000
PLS M1000
Change the open command to pulse
Designate the control data for the OPEN instruction
(It is necessary to designate either one of the processing enclosed by a broken line.)
1)
M1000 X19 M0 M20
MOVP H0 D100
When using the open setting parameters of GX Developer
Designate the execution type/complete type
2)
When not using the open setting parameters of GX Developer
MOVP H8000 D100
Designate the execution type/complete type
MOVP H0 D102
Designate the information of the application
MOVP H1000 D103
Designate the local station Port No. (1000H)
DMOVP H0A6155DF D104
Designate the external device IP address (10.97.85.223)
MOVP H2000 D106
Designate the external device Port No. (2000H)
ZP.OPEN "U0" K1 D100 M100
Execute the open processing to the external device
M100 M101
SET M150
Set the normal completion flag of the open processing
M101
SET M151
Set the abnormal completion flag of the open processing

* Regarding sections 1) and 2) in the program, 1) is necessary when using the GX Developer "Open Setting" parameter. 2) is necessary when not using the GX Developer "Open Setting" parameter.

10.9 UINI Instruction

This instruction is used to perform the re-initial processing of the Ethernet module.

Setting data	Applicable device									
	Internal device (System, user)		File register	MELSECNET/H, MELSECNET/10 Direct J□\□		Special module U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	$	
(S1)	–	○		–				–	–	–
(D1)	○	○		–				–	–	–



Setting data

Setting data	Description	Set by (*1)	Data type
"Un"	Start input/output signal of the Ethernet module (00 to FE: The two most significant digits of the 3-digit input/output signal)	User	Binary 16 bits
(S1)	Head number of the device that stores control data	User, system	Binary 16 bits
(D1)	Head number of the local station bit device that turns on for one scan upon completion of instruction. (D1) + 1 also turns on if the instruction execution ends abnormally.	System	Bit

The file registers for each of the local device and the program cannot be used as devices to be used in the setting data.

10 - 23

Control data

Device	Item	Setting data	Setting range	Set by (*1)
(S1) + 0	System area	—	—	—
(S1) + 1	Completion status	• Stores the status at completion. 0000H: Normal completion Other than 0000H : Abnormal completion (error code) (*2)	—	System
(S1) + 2	Specification of target of change	• Specifies the parameters to be changed. b15 to b2 b1 b0 0 | 2) | 1) 1) Specification of whether or not to change the IP address of the local station Specifies whether or not to change the IP address of the local station. (Specifies the address using (S1) + 3 and (S1) + 4 if it is to be changed.) 0: Do not change 1: Change 2) Specification of whether or not to change the operation settings Specifies whether or not to change the operation settings. (Specifies the settings using (S1) + 5 if it is to be changed.) 0: Do not change 1: Change	0H to 3H	User
(S1) + 3 (S1) + 4	Local station IP address	• Specifies the IP address of the local station.	00000001H to FFFFFFFEH	User
(S1) + 5	Operation settings (See Section 4.7.)	• Specifies the operation settings. b15 to b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 0 | 5) | 0 | 4) | 3) | 2) | 0 | 1) | 0 1) Communication data code setting 0: Communication in binary code 1: Communication in ASCII code 2) TCP Existence confirmation setting 0: Use the Ping 1: Use the KeepAlive 3) Send frame setting 0: Ethernet frame 1: IEEE802.3 frame 4) Setting of write enable/disable at RUN time 0: Disable 1: Enable 5) Initial timing setting 0: Do not wait for OPEN (communication impossible at STOP time) 1: Always wait for OPEN (communication possible at STOP time)	(See the description to the left.)	User

*1 The "Set by" column indicates the following:
- User : Data set by the user before executing a dedicated instruction.
- System : The PLC CPU stores the execution results of a dedicated instruction.

*2 For details on the error codes at abnormal completion, see Section 11.3, "Error Code List".

POINT

When performing re-initial processing of the Ethernet module only, i.e., without changing the local station IP address and operation settings, the control data should be specified so that the value (0H) is stored in ((S1) + 2)), the specification of target of change, before executing the UINI instruction.
The Ethernet module clears external device address information that it has been maintaining and performs re-initial processing in order to allow data communication to restart. (The initial normal completion signal (X19) is on.)

Functions

(1) Perform the re-initial processing of the Ethernet module specified in Un.

(2) Whether or not the UINI instruction has been completed can be checked by the complete bit devices (D1) + 0 and (D1) + 1.

(a) Complete bit device (D1) + 0

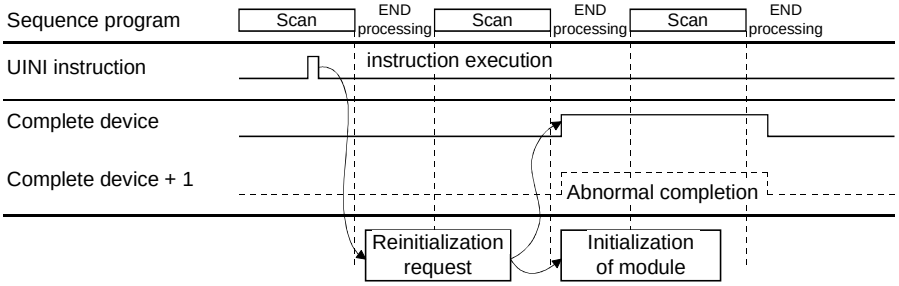
Turns on at the END processing of the scan where the UINI instruction is completed, and turns off at the next END processing.

(b) Complete bit device (D1) + 1

Turns on and off depending on the completion status of the UINI instruction.

- Normal completion : Stays off and does not change.
- Abnormal completion : Turns on at the END processing of the scan where the UINI instruction is completed, and turns off at the next END processing.

[Operation when the UINI instruction is being executed]



(3) The ZP.UINI is executed when the open instruction switches from off to on.

POINT

Please keep the following points in mind when reinitializing Ethernet module. (Failure to do so may cause errors in the data communication with the external devices.)

(1) Be sure to end all current data communication with external devices and close all connections before performing a re-initial processing.

(2) Do not mix a re-initial processing done by writing directly into buffer memory, for instance by using a TO instruction, with a re-initial processing via UINI instruction. Also, do not request another re-initial processing while a re-initial processing is already being performed.

(3) Be sure to reset external devices if the IP address of the Ethernet module has been changed. (If an external device maintains the Ethernet address of a device with which it communicates, the communication may not be continued after the IP address of the Ethernet module has been changed.)

(4) In a redundant system, do not change the IP address and operation setting with the UINI instruction. If they are changed, normal communication cannot be made.
Use GX Developer to change the IP address and operation setting.

Errors

(1) When a dedicated instruction ends with an error, the abnormal completion signal, (D1) + 1, turns on and the error code is stored in the complete status area (S1) + 1. Refer to the following manuals regarding the error code, check the errors and take corrective actions.

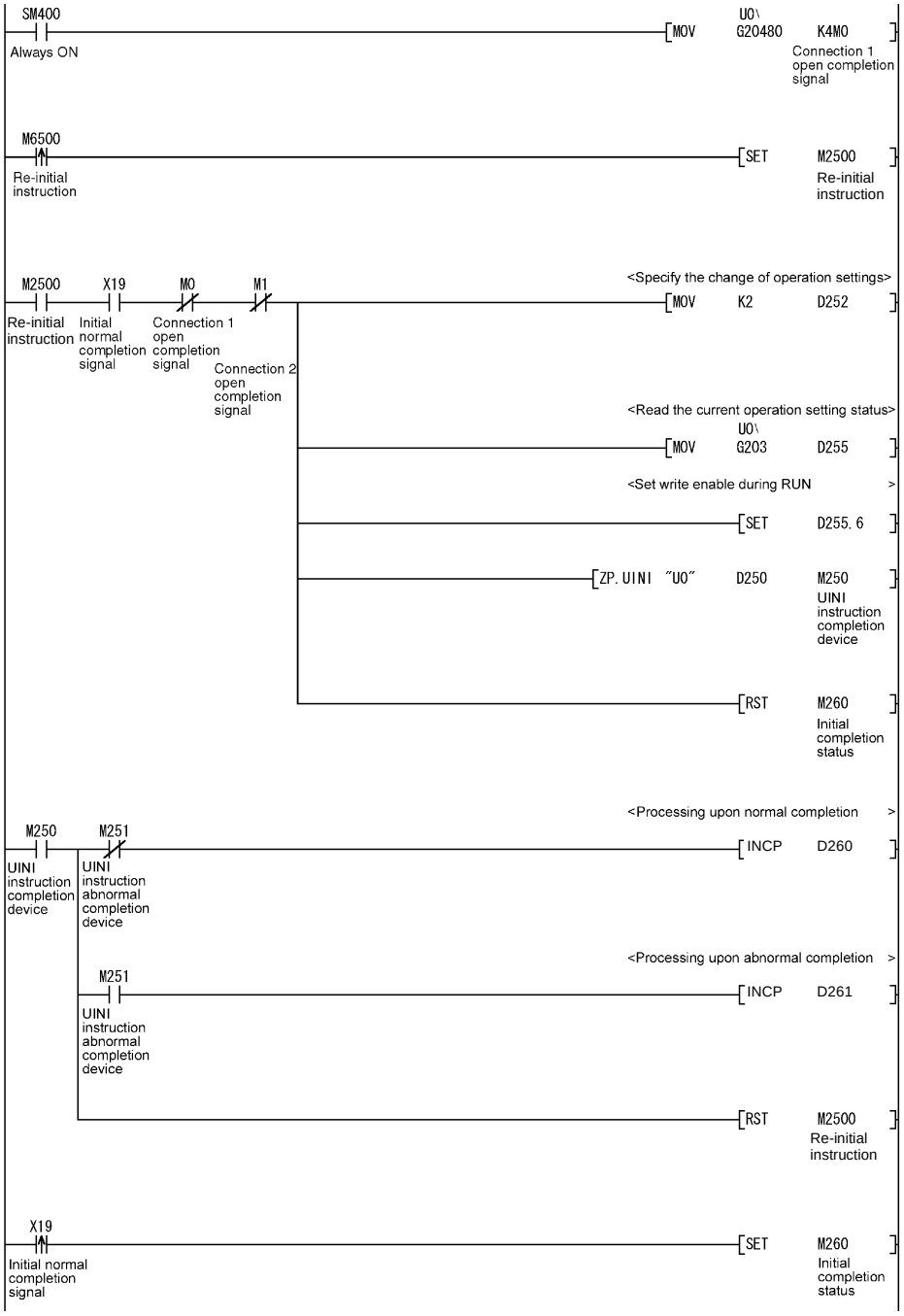
<Error codes>

4FFF$_H$ or less : QCPU (Q Mode) User's Manual (Hardware Design, Maintenance and Inspection)

C000$_H$ or higher : Section 11.3.3 of this manual

Program example

The following figure shows a sample program that performs a re-initial processing.
When I/O signals of the Ethernet module are X/Y00 to X/Y1F



POINT
This is an example program that can be used to perform re-initial processing when communicating with connections 1 and 2. The corresponding numbers and bits for each connection should be specified if other connections are used.

11 TROUBLESHOOTING

This section explains the contents of the errors that may occur during communications between the Ethernet module and an external device as well as the troubleshooting procedures.
The following are methods for checking whether there is an error on the Ethernet module side and the contents of the error.
Use one of the following methods to check whether there is an error and its content, then take corrective actions.

(1) Check using the display LED on the front of the Ethernet module (See Section 11.1.)

The display LED on/off status can be used to check whether an error is occurring in the Ethernet module.

(2) Check through GX Developer

GX Developer can be used to check various conditions of the Ethernet module as well as the error code corresponding to the contents of the error occurring and to perform tests.

(a) Ethernet diagnostics (using the dedicated screen)

1) Monitor for the status of various settings (See Section 11.2.)
2) PING test (See Section 5.4.1.)
3) Loop back test (See Section 5.4.2.)
4) COM. ERR off (See Sections 11.1.2 and 11.2.1.)

(b) System monitor (using the dedicated screen: see Section 11.2.2.)

1) Module's detailed information · · · · Module status, error code, etc.
2) H/W information · · · · LED on/off status, switch status, etc.

(c) Buffer memory batch monitor

The error code can be checked by monitoring the buffer memory of the Ethernet module.

(3) Check the contents of the error using the error code (See Section 11.3.)

The contents of the error can be checked using the error code confirmed on the above dedicated screen or by monitoring the buffer memory by referring to Section 11.3.

REMARK

If line errors and other errors occur when connecting devices of multiple manufacturers, the users needs to isolate the malfunctioning parts using line analyzers, etc.

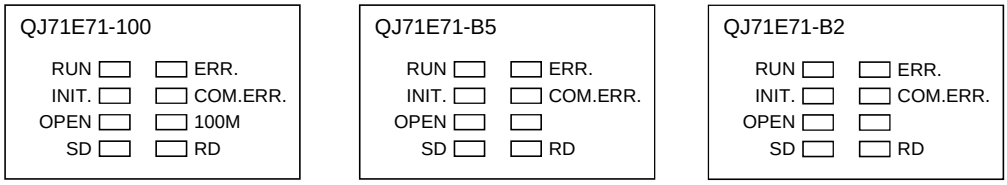
11.1 How to Check Errors Using LED Displays

This section describes the errors that can be checked with the LED displays on the front of the Ethernet module.

11.1.1 Checking error display

The following can be checked with the LED displays on the front of the Ethernet module.

<Ethernet module LED>



	LED name	Status to check	Cause/corrective action
1	[RUN]	Turns off after powering on the Ethernet module. (*1)	1) Watchdog timer error • When a watchdog timer (approximately 600 ms) error occurs, the watchdog timer error detection signal (X1F) is turned on by the self diagnosis function of the Ethernet module. 2) Ethernet module installation fault • Check if the power supply capacity (5 V DC) of the power supply module is insufficient. • Turn off the power supply and reinstall the module.
2	[ERR.]	Turns on after powering on the Ethernet module. (*1)	1) Module parameter setting error • Check/correct the parameter setting values for the Ethernet module using GX Developer. 2) PLC CPU error • When the PLC CPU's [RUN] LED is off/flashing, or the [ERR.] LED is on, check the content of the error occurring in the PLC CPU and correct the problem. • Check that the Ethernet module is installed on the Q mode PLC CPU. 3) Ethernet module error (H/W error)
3	[COM.ERR]	Turns on after powering on the Ethernet module.	1) Check the contents of the error using the error codes stored by the error detection of the following processing and remove the causes. • Initial processing • Fixed buffer send processing • E-mail send/receive processing • Open processing • Data communication processing • Other processing (processing for which error codes are stored in the error log area) 2) For a list of error codes, see Section 11.3.

(Continues on the following page)

(Continued from the previous page)

	LED name	Status to check	Cause/corrective action
4	[SD]	The [SD] LED does not flash at data sending.	1) [ERR.] or [COM.ERR] LED turns on. • Remove the factors that turn on the [ERR.] or [COM.ERR] LED. 2) Poor cable connection • Check the connection of the cable. (∗2) 3) Program reviewing is required • Review the sequence program for sending.
5	[RD]	[RD] LED stays off and data cannot be received.	1) [ERR.] or [COM.ERR] LED turns on. • Remove the factors that turn on the [ERR.] or [COM.ERR] LED. 2) Poor cable connection • Check the connection of the cable. (∗2) 3) Local station IP address setting error • If the cable connection is all right, review each setting value of the local station IP address, router setting, and sub-net mask settings using GX Developer. 4) Program reviewing is required • Review the sequence program for sending.

∗1 Conduct a hardware test (H/W test) and check whether or not the Ethernet module operates normally.
For details on the hardware test, see Section 4.8.2, "Hardware test".

∗2 Conduct a confirmation of the completion for the initial processing and check whether or not there is any problem in the cable connection and the Ethernet lines.
See Section 5.4, "Confirming the completion of the Initial Processing" for details on confirming the completion for the initial processing. (Perform either one of the "Confirming the completion of the initial processing completion" actions described in Section 5.4.)

POINT

The on/off status of the [INIT], [OPEN], [ERR.] and [COM.ERR] LEDs is stored in the module status area (address: C8H) of the buffer memory.
For more details, see Section 3.8, "List of Applications and Assignments of the Buffer Memory".

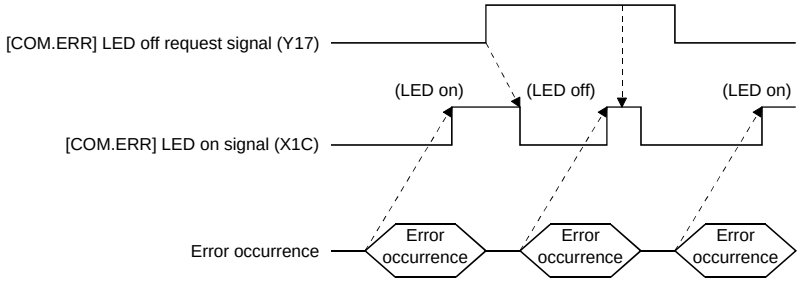
11.1.2 How to turn off COM.ERR LED and to read/clear error information

This section explains how to turn off [COM.ERR] LED and to read/clear error information using the sequence program.

(1) How to turn off [COM.ERR] LED using input/output signals

The [COM.ERR] LED on the front of the Ethernet module is turned on when a communication error occurs in an external device. (Input/output signal X1C: ON)

(a) The [COM.ERR] LED is turned off by turning on the off request signal (Y17).



(b) The off request is processed continuously while the off request signal (Y17) is on.

(c) The error information in the error log area of the buffer memory is not cleared (deleted) by turning the off request signal (Y17) is on.

(2) How to turn off [COM.ERR] LED on the "Ethernet diagnostics" screen of GX Developer (See Section 11.2.1.)

(a) Clicking on the COM.ERR off button turns the [COM.ERR] LED off.

(b) The error information in the error log area of the buffer memory is not cleared (deleted).

(3) How to read/clear error information using the dedicated instructions

Error information can be read/cleared at arbitrary timing by using the following dedicated instructions.

(a) Dedicated ERRRD instruction
Using this instruction, initial abnormal code information or open abnormal code information can be read.

(b) Dedicated ERRCLR instruction
Using this instruction, it is possible to turn off [COM.ERR]LED and clear initial abnormal code/open abnormal code, or error log.

* For the details of the dedicated instruction, see Chapter 10, "Dedicated Instructions".

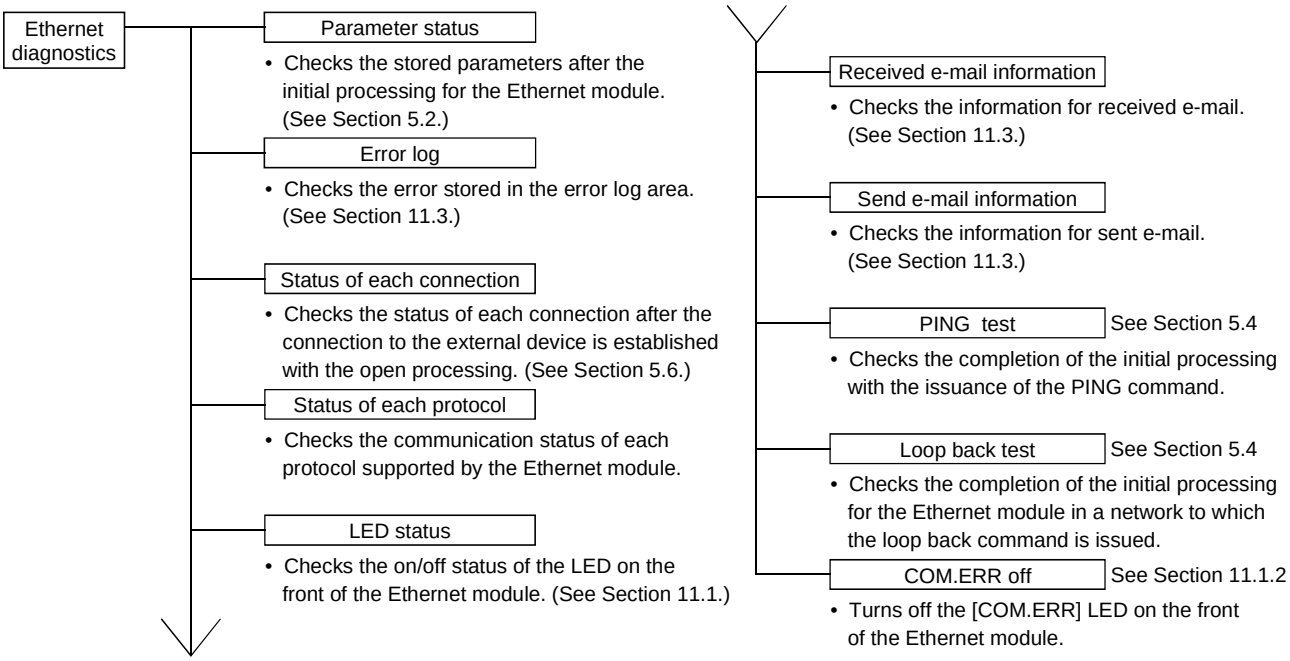
11.2 How to Check an Error Through GX Developer

The status of the various settings for the Ethernet module can be checked using the GX Developer functions.

(1) Ethernet diagnostics (See Section 11.2.1.)

The module status of an Ethernet module, parameter settings, communication status, error log and others can be checked using the Ethernet diagnostic function.

The following are the functions of the Ethernet diagnostics.


Ethernet diagnostics
Parameter status
• Checks the stored parameters after the initial processing for the Ethernet module. (See Section 5.2.)
Error log
• Checks the error stored in the error log area. (See Section 11.3.)
Status of each connection
• Checks the status of each connection after the connection to the external device is established with the open processing. (See Section 5.6.)
Status of each protocol
• Checks the communication status of each protocol supported by the Ethernet module.
LED status
• Checks the on/off status of the LED on the front of the Ethernet module. (See Section 11.1.)
Received e-mail information
• Checks the information for received e-mail. (See Section 11.3.)
Send e-mail information
• Checks the information for sent e-mail. (See Section 11.3.)
PING test
See Section 5.4
• Checks the completion of the initial processing with the issuance of the PING command.
Loop back test
See Section 5.4
• Checks the completion of the initial processing for the Ethernet module in a network to which the loop back command is issued.
COM.ERR off
See Section 11.1.2
• Turns off the [COM.ERR] LED on the front of the Ethernet module.

(2) System monitor (See Section 11.2.2.)

The module status of an Ethernet module can be checked from the system monitor.

(a) Module's detailed information

The function version and error code can be checked.

(b) H/W information

LED on/off status, connection status and parameter status of an Ethernet module can be checked.

(3) Buffer memory batch monitor (See Section 11.2.4.)

The buffer memory of an Ethernet module is monitored.

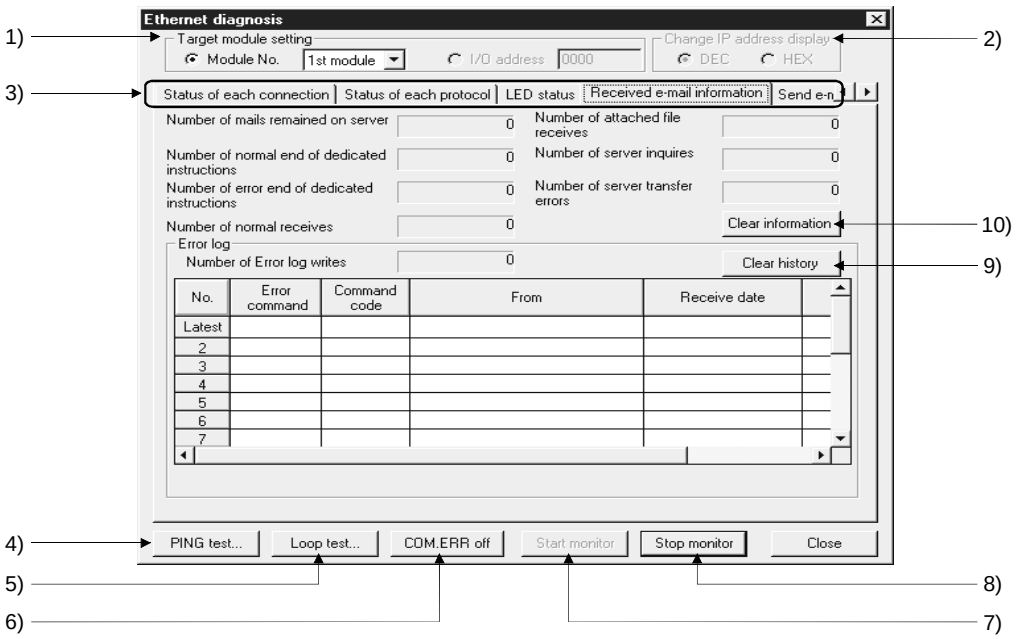
POINT
See Section 11.2.3 for the buffer memory that can be checked on the "Ethernet diagnostics" screen and "System monitor" screen.

11.2.1 Ethernet diagnostics

[Purpose]
The module status of an Ethernet module, parameter settings, communication status, error log and others can be checked using the Ethernet diagnostic function of GX Developer.

[Operating procedure]
GX Developer → [Diagnostics] → Ethernet diagnostics

[Ethernet diagnostics screen]



[Explanation of items]

No.	Item	Description	Setting range
1	Target module setting	Specifies the target Ethernet module for monitoring. ＊ Number of cards for MELSECNET/H module is not included.	Card 1 to Card 4
2	Change IP address display	Switches the IP address display between decimal and hexadecimal.	Decimal/hexadecimal
3	Selection from the various information monitors	Various types of information for the Ethernet module can be monitored. (See Section 11.2.3 for the buffer memory corresponding to the display contents.)	—
4	PING test	Performs the PING test on the external device. (See Section 5.4.1, Section 5.4.2.)	
5	Loop test	Performs the loop back test for the network. (See Section 5.4.3.)	
6	COM. ERR off	Clicking this button turns the [COM.ERR] LED off. (See Section 11.1.2.)	
7	Start monitor	Clicking this button executes the Ethernet diagnostics. The display is updated during monitoring.	
8	Stop monitor	Clicking this button stops the Ethernet diagnostics. The display is retained while monitoring is stopped.	
9	Clear history	Clears the log.	
10	Clear information	Clears various values.	

POINT

When accessing a PLC on other station with a data link instruction during Ethernet diagnostics, it may take some time to execute the data instruction.
Take the following actions and then execute the Ethernet diagnostics when executing data link instructions.

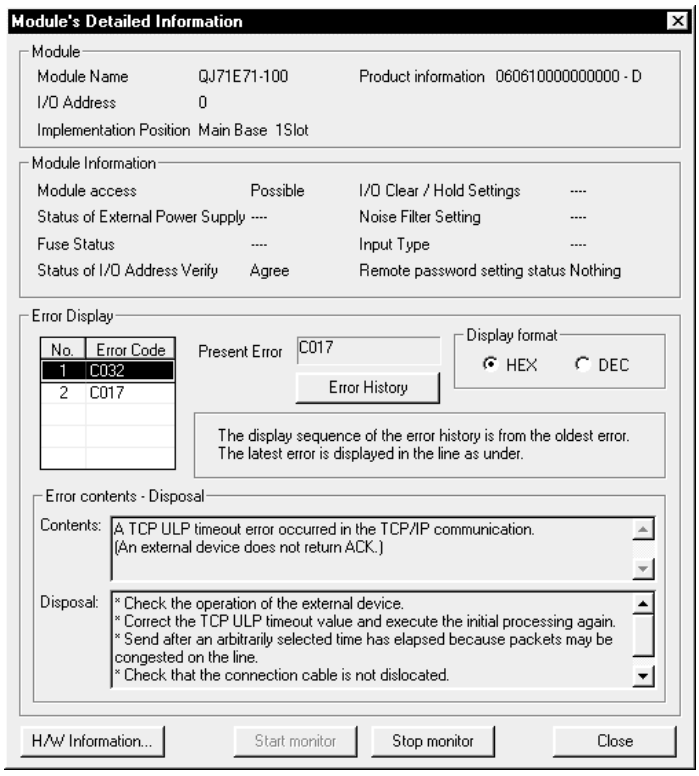
- Execute the COM instruction.
- Using the special register SD315, ensure the time for communication processing of 2 to 3 ms.

11.2.2 System monitor

The module status of the Ethernet module can be checked from the system monitor.

(1) Checking the module status and error codes on the detailed module information screen for the diagnostic functions.

[Startup procedure]
GX Developer → [Diagnostics] → [System monitor] →
Module's Detailed Information


Module's Detailed Information
Module
Module Name QJ71E71-100
Product information 060610000000000 - D
I/O Address 0
Implementation Position Main Base 1Slot
Module Information
Module access Possible
I/O Clear / Hold Settings ----
Status of External Power Supply ----
Noise Filter Setting ----
Fuse Status ----
Input Type ----
Status of I/O Address Verify Agree
Remote password setting status Nothing
Error Display
No. Error Code
1 C032
2 C017
Present Error C017
Error History
Display format
HEX DEC
The display sequence of the error history is from the oldest error.
The latest error is displayed in the line as under.
Error contents - Disposal
Contents: A TCP ULP timeout error occurred in the TCP/IP communication.
(An external device does not return ACK.)
Disposal: * Check the operation of the external device.
* Correct the TCP ULP timeout value and execute the initial processing again.
* Send after an arbitrarily selected time has elapsed because packets may be congested on the line.
* Check that the connection cable is not dislocated.
H/W Information...
Start monitor
Stop monitor
Close

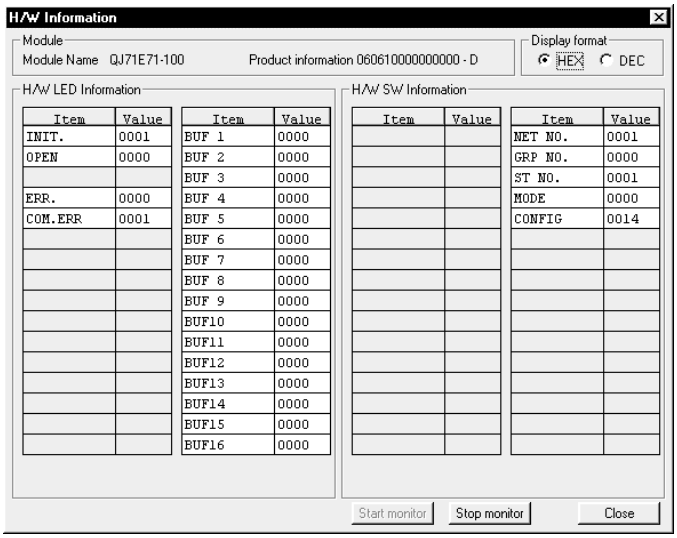
[Displays]

- Module
  The following information is displayed:
  Model name : the model name of the module installed
  Starting I/O No. : the starting input/output signal number of the target module
  Mounting position : the slot position where the module is mounted
  Product information : Product information
  ＊ The function version of the module is shown at the end of the product information.
  (Example) A "B" at the end of the product information indicates this is a module of function version B.
- Unit access (Module access)
  Displays access permissions when the WDT error signal (X1F) is turned off.
- Status of I/O Address Verify
  Displays whether or not the module for which the user has set the parameters matches the module installed.
- Remote password setting status
  Displays the remote password setting status.
- Present Error
  Displays the error code of the latest error occurred.

- Error Display
  Displays the error codes stored in the error log area (address: E5H, EEH …, 16CH) of the buffer memory.
- Error details, corrective action
  Displays the error details and corrective action for the error code selected in Error Display.

(2) Checking the LED on/off status and operation mode number on the H/W information screen of the diagnostic functions

[Startup procedure]
GX Developer → [Diagnostics] → [System monitor] →
Module's Detailed Information → H/W Information



[Displays]
This screen displays the Ethernet module information stored in the following areas of the buffer memory.

No.	Display content		Corresponding buffer memory	Address
1	H/W LED information left side (*1)		Storage area for LEDs on station	C8H (200)
2	H/W LED information right side		Storage area for connection status	5000H (20480)
1	H/W switch information	Network No.	Local station network number/station number storage area	76H (118)
2		Group No.	Local station group number storage area	77H (119)
3		Station No.	Local station network number/station number storage area	76H (118)
4		Mode No.	Switch status (operation mode setting) storage area	CAH (202)
5		Communication status (*2)	Communication status storage area	CBH (203)

*1

b15 to b5	b4	b3	b2	b1	b0
–	1/0	1/0	–	1/0	1/0

b4: COM.ERR LED, b3: ERR.LED, b1: OPEN LED, b0: INIT.LED

1: On
0: Off

*2

b15 to b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
–	1/0	–	1/0	1/0	1/0	–	1/0	1/0	–

6) 5) 4) 3) 2) 1)

1) Communication data code
   0: Binary
   1: ASCII
2) Initial/Open method
   0: No parameter setting
   1: With parameter setting
3) TCP existence confirmation
   0: Ping
   1: KeepAlive
4) Send frame
   0: Ethernet
   1: IEEE802.3
5) Write enable/disable at RUN time
   0: Disable
   1: Enable
6) Initial timing
   0: Do not wait for OPEN
   1: Always wait for OPEN

11.2.3 Buffer memory that can be monitored with the GX Developer diagnostic function

The following is a list of the buffer memory addresses that can be displayed on the "Ethernet diagnostics" screen and "System monitor" screen of GX Developer.
The display screen and display contents are mentioned for the Ethernet diagnostics screen. See Section 11.2.2 for how to display the System monitor screen.

Address Dec (Hex)	Applicable buffer memory		Ethernet diagnostics display screen	Display contents	
32 (20H)	Connection No. 1	Destination existence confirmation (b1)	Status for each connection	Connection No. 1	Existence confirmation
		Pairing open (b7)			Pairing open
		Communication method (protocol) (b8)			Protocol
		Open system (b15, b14)			Open system
33 to 39 (21H to 27H)	Connection No. 2 to 8 (same as connection No. 1)			Connection No. 2 to 8	
105 (69H)	Initial error code		Parameter status	Module information	Initial error code
106 to 107 (6AH to 6BH)	Local station IP address				IP address
108 to 110 (6CH to 6EH)	Local station Ethernet address				Ethernet address
116 (74H)	Automatic open UDP port number				Automatic open UDP port number
118 (76H) *1	Local station network number/station number				Network No.
					Station No.
119 (77H) *1	Local station group number				Group No.
120 (78H)	Connection No. 1	Local station Port No.	Status of each connection	Connection No. 1	Local station Port No.
121 to 122 (79H to 7AH)		Destination IP address			Destination IP address
123 (7BH)		Destination Port No.			Destination Port No.
124 (7CH)		Open error code			Open error code
125 (7DH)		Fixed buffer sending error code			Fixed buffer transfer error code
126 (7EH)		Connection end code			Connection end code
130 to 199 (82H to C7H)	Connection No. 2 to 8 (same as connection No. 1)			Connection No. 2 to 8	
200 (C8H) *1	LED on/off status	[INIT.]LED (b0)	LED status	LED display status	INIT.
		[OPEN]LED (b1)			OPEN
		[ERR.]LED (b3)			ERR.
		[COM.ERR] LED (b4)			COM.ERR
202 (CAH) *1	Switch status (operational mode setting)			Drive mode	
203 (CBH) *1	Status of setting with GX Developer		—	(Check on the System monitor screen.)	
227 (E3H)	Error log block 1	Number of error occurrence	Error log	Latest	Number of error occurrence
229 (E5H)		Error code/end code			Error code/end code
230 (E6H)		Subheader			Subheader
231 (E7H)		Command code			Command code
232 (E8H)		Connection No.			Connection No.
233 (E9H)		Local station Port No.			Local station Port No.
234 to 235 (EAH to EBH)		Destination IP address			Destination IP address
236 (ECH)		Destination Port No.			Destination Port No.
238 to 372 (EEH to 174H)	Error log block 2 to 16 (same as error log block 1)			No. 2 to No. 16	
376 to 377 (178H to 179H)	IP	Received IP packet count	Status for each protocol	IP packet	Total number of receives
378 to 379 (17AH to 17BH)		Received IP packet count discarded due to sum check error			Total number of Sum check error annulments
380 to 381 (17CH to 17DH)		Sent IP packet total count			Total number of sends
408 to 409 (198H to 199H)	ICMP	Received ICMP packet count		ICMP packet	Total number of receives
410 to 411 (19AH to 19BH)		Received ICMP packet count discarded due to sum check error			Total number of Sum check error annulments
412 to 413 (19CH to 19DH)		Sent ICMP packet total count			Total number of sends
414 to 415 (19EH to 19FH)		Echo request total count of received ICMP packets			Total number of echo request receives
416 to 417 (1A0H to 1A1H)		Echo reply total count of sent ICMP packets			Total number of echo reply sends
418 to 419 (1A2H to 1A3H)		Echo request total count of sent ICMP packets			Total number of echo request sends

Address Decimal (hexadecimal)	Applicable buffer memory			Ethernet diagnostics display screen	Display contents	
420 to 421 (1A4H to 1A5H)	ICMP	Echo reply total count of received ICMP packets		Status of each protocol	ICMP packet	Total number of echo reply receives
440 to 441 (1B8H to 1B9H)	TCP	Received TCP packet count			TCP packet	Total number of receives
442 to 443 (1BAH to 1BBH)		Received TCP packet count discarded due to sum check error				Total number of Sum check error annulments
444 to 445 (1BCH to 1BDH)		Sent TCP packet total count				Total number of sends
472 to 473 (1D8H to 1D9H)	UDP	Received UDP packet total count			UDP packet	Total number of receives
474 to 475 (1DAH to 1DBH)		Received UDP packet count discarded due to sum check error				Total number of Sum check error annulments
476 to 477 (1DCH to 1DDH)		Sent UDP packet total count				Total number of sends
20480 (5000H) ✻1	Open completed signal for connections No. 1 to No. 16			—	(Check on the System monitor screen.)	
22560 to 22639 (5820H to 586FH)	Connection No. 9 to 16 (same as connection No. 1 (address: 120 to 126 (78H to 7EH)))			status of each connection	Connection No. 9 to 16	
22640 (5870H)	Receive	Number of mail remaining on the server		Received e-mail information	Number of mail remained on the server	
22641 (5871H)		Dedicated instruction normal completion count			Number of normal end of dedicated instructions	
22642 (5872H)		Dedicated instruction abnormal completion count			Number of error end of dedicated instructions	
22643 (5873H)		Normal receiving count			Number of normal receives	
22644 (5874H)		Attached file receiving count			Number of attached file receives	
22645 (5875H)		Server inquiry count			Number of sever inquires	
22646 (5876H)		Server communication error count			Number of server transfer errors	
22647 (5877H)		Error log write count			Number of Error log writes	
22649 (5879H)		Error log block 1	Error code		Latest error log	Error code
22650 (587AH)			Command code			Command code
22651 to 22658 (587BH to 5882H)			From			From
22659 to 22662 (5883H to 5886H)			Date			Receive date
22663 to 22692 (5887H to 58A4H)			Subject			Subject
22693 to 23352 (58A5H to 5B38H)		Error log block 2 to 16 (same as error log block 1)			Error log 2 to 16	
23353 (5B39H)	Send	Dedicated instruction normal completion count		Sent e-mail information	Number of normal end dedicated instructions	
23354 (5B3AH)		Dedicated instruction abnormal completion count			Number of error end of dedicated instructions	
23355 (5B3BH)		Number of mails normally completed			Number of normal end mails	
23356 (5B3CH)		Attached file sending count			Number of attached file sends	
23357 (5B3DH)		Sending to the server count			Number of server sends	
23358 (5B3EH)		Number of mails abnormally completed			Number of error end mails	
23359 (5B3FH)		Error log write count			Number of Error log writes	
23361 (5B41H)		Error log block 1	Error code		Latest error log	Error code
23362 (5B42H)			Command code			Command code
23363 to 23370 (5B43H to 5B4AH)			To			Send To
23371 to 23374 (5B4BH to 5B4EH)			Date			Send date
23375 to 23404 (5B4FH to 5B6CH)			Subject			Subject
23405 to 23712 (5B6DH to 5CA0H)		Error log block 2 to 16 (same as error log block 1)			Error log 2 to 16	

✻1 Can be monitored on the System monitor screen of GX Developer. See Section 11.2.2.

11 - 11

11.2.4 Checking the error information by the buffer memory batch monitoring function

It is explained here how the Ethernet module errors can be checked from GX Developer.
Error codes stored in the buffer memory of the Ethernet module can be monitored using the "Buffer memory batch monitoring" function of GX Developer.

[Operating procedure]

(Step 1) Select [Online] – [Monitor] – [Buffer memory batch] from the GX Developer menu bar, and start the "Buffer memory batch monitoring" screen.

(Step 2) Enter [Module start address:].
For the module start address, enter the start I/O signal of the Ethernet module to be monitored using four digits.
(Example)
Start I/O signal: Enter "0020" when monitoring modules of X/Y0020 to 003F

(Step 3) Enter [Buffer memory start address:].
Enter the buffer memory address to be monitored, using the selected input format (decimal/hexadecimal).
For a list of the buffer memory addresses where error codes are stored, see Section 11.3, "Error Code List".
(Example)
When monitoring the initial abnormal code (buffer memory storage address: 69H):
Enter "69" + "hexadecimal"

(Step 4) Click the Start Monitor button.
The contents of the buffer memory after the specified address are displayed. (In case of the above example, the contents of 69H and succeeding addresses are displayed.)

REMARK

The display format can by modified as follows:
Monitor format : Bits & words/ Multiple bit points/ Multiple word points
Display : 16-bit integer/32-bit integer/real number/ASCII character
Numerical value : Decimal/hexadecimal

For details, refer to the "Operating Manual" for GX Developer.

11.3 Error Code List

This section explains the error codes (abnormal codes) for the errors that may occur in each processing when communicating data between the Ethernet module and an external device as well as those generated by processing requests from the local station's QCPU.
The details of errors and error handling procedures are described.

	Type of error	Description	Error code storage buffer memory	Explanation
1	Errors occurring in initial processing	• Setting value error • Initial processing error	69H: Initial error code (Communication status storage area)	Section 11.3.3
2	Errors occurring in open processing	• Setting value error • Open processing error	7CH: 5824H: Open error code (Communication status storage area)	
3	Errors occurring in fixed buffer sending to an external device	• Designated data error • Sending error	7DH: 5825H: Fixed buffer sending error code 7EH: 5826H: Connection end code (Communication status storage area)	
4	Errors occurring in fixed buffer communication with an external device	• Designated data error • Communication error (exclude 3 above)	7EH: 5826H: Connection end code (Communication status storage area)	
5	Errors returned to an external device when communicating with the external device	• Errors returned in fixed buffer communication (end code)		Section 11.3.1
		• Errors returned in random access buffer communication (end code)		
		• Errors returned in communication using the MC protocol	End codes when QnA compatible 3E frame commands are used	Section 11.3.3
			End codes when A compatible 1E frame commands are used	Section 11.3.1
			Error codes when A compatible 1E frame commands are used	Section 11.3.2
6	Errors occurring while communicating with an external device (including the causes shown in the description column), and whose error codes are stored in the error log area.	• Designated data errors • Errors whose source cannot be confirmed • Errors that occurred while communicating using the random access buffer • Errors that occurred while communicating using the MC protocol	E5H: (Error log area)	Section 11.3.3
7	Errors (response commands) occurring in communication with an external device using file transfer (FTP) function	• Designated data error • Communication error, etc.		CPU module manual
8	Error ocurring when communicating with the Web function	• Communication error	5101H··· (HTTP status stroage area)	Section 11.3.3
9	Errors occurring when receiving e-mail	• Setting data error • Receiving error	5870H: Receive (E-mail status storage area)	
10	Errors occurring when sending e-mail	• Setting data error • Sending error	5B39H: Send (E-mail status storage area)	
11	Errors occurring in communication using data link instructions from the local station's QCPU	• Designated data error • Communication error	Not stored (Stored in the complete status area of the instruction)	
12	Errors occurring in communication using dedicated instructions from the local station's QCPU	• Designated data error • Communication data error	Not stored (Stored in the complete status area of the instruction)	

* For details on the error codes returned by MX Component when communicating data using MX Component, refer to the MX Component Programming Manual.

(1) Initial error code (address: 69H)

(a) This address stores the error codes generated when the initial processing is executed.

(b) Error codes are stored as binary values when the initial abnormal completion signal (X1A) is on.

(c) An error code is cleared when the initial normal completion signal (X19) is turns on, but can also be cleared by the following operations.

1) Resetting the PLC CPU or turning off the PLC power

2) Writing "0" to the initial error code storage area with a sequence program

(2) Open error code
(connection numbers: 1 to 8; addresses: 7CH to C1H)
(connection numbers: 9 to 16; addresses: 5824H to 5869H)

(a) These addresses store the result of the open processing for the applicable connections.

(b) The results of the open processing are stored in binary values.

0	: Normal completion
Other than 0	: Abnormal completion (Open abnormal detection signal (X18): ON)

(c) An error code is cleared by the following operations.

1) Reopening the connection that caused an open error

2) Resetting the PLC CPU or turning off the PLC power

(3) Fixed buffer sending error code
(connection numbers: 1 to 8; addresses: 7DH to C2H)
(connection numbers: 9 to 16; addresses: 5825H to 586AH)

(a) These addresses store error codes generated when an error in data sending to an external device occurs during fixed buffer communication using the applicable connection.

(b) A sending error code is cleared when the next data sending is normally completed.

(4) Connection end code
(connection numbers: 1 to 8; addresses: 7EH to C3H)
(connection numbers: 9 to 16; addresses: 5826H to 586BH)

(a) These addresses store end codes returned from an external device as a response during the fixed buffer communication using the applicable connection.

(b) Determine how to handle the end codes in the responses by arranging with the particular external device.

(5) Error log area (address: E0H to 1FFH)

This area stores the following errors.

- Errors whose source cannot be confirmed
- Errors that occurred during communications using the random access buffer
- Errors that occurred during communications using the MC protocol

(a) Number of error occurrences (address: E3H)

1) This address stores the number of errors registered in the error log block area.
2) When errors occur more than 65536 times, the count is stopped at FFFFH (65535).

(b) Error log write pointer (address: E4H)

1) This address stores the error log block number where the latest error logging is registered.
   0 : No error. (No registration of error log)
   1 or more : Error log block number where the latest error logging is registered
   ∗ If the pointer value is "16", it means that the latest error logging is registered in the error log block 16 area.
2) When 17 or more errors occur, the registration of error logging starts from error log block 1 area again.

POINT

(1) An error log block area consists of sixteen error log blocks that have the same data order.
(2) The error information is continued to be stored into the following areas even if the count of the error occurrences is stopped and no longer stored:
- Error log write pointer storage area
- Error log block

(c) Error log block – Error code/end code (address: Starting from E5H)

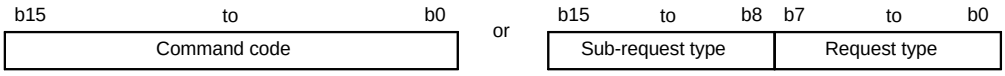
1) In the error code/end code area, the error codes that indicate error content are stored. (See Section 11.3.)

(d) Error block – Subheader (address: starting from E6H)

1) The subheader code of a faulty message is stored in bits 0 to 7 of the corresponding area. ("0" is stored in bits 8 to 15).
2) For errors below the TCP and UDP levels, "0" is stored.

(e) Error log block – Command code (address: starting from E7H)

1) This block stores the command code of a faulty message, or each lower byte value of request type and sub-request type of a data link instruction.



2) "0" is stored in the following case.
- For messages not containing a command code
- For errors below the TCP and UDP levels (because their commands are unknown)

(f) Error log block – Connection No. (address: starting from E8H)

1) The faulty connection No. is stored in bits 0 to 7 of the corresponding area. ("0" is stored in bits 8 to 15).

2) For errors below the TCP and UDP levels, "0" is stored.

(g) Error log block – Local station Port No. (address: starting from E9H)

1) This block stores the local station's port No. when an error occurred.

2) "0" is stored for errors below the TCP and UDP levels.

(h) Error log block – Destination IP address (address: EAH and starting from EBH)

1) This block stores the IP address of an external device when an error occurred.

2) "0" is stored in the following cases.
- For errors below the IP level
- When an error response was performed by relaying through the PLC CPU.

(i) Error log block – Destination Port No. (address: starting from ECH)

1) This block stores the external device's port No. when an error occurred.

2) "0" is stored for errors below the TCP and UDP levels.

(j) Status for each protocol (addresses: 178H to 1FFH)

1) This address stores the count of the occurrence for the applicable contents of each protocol's condition.
(The value counted by the Ethernet module.)

2) When the count value exceeds two words, the count stops at FFFFFFFFH (4294967295).

POINT

Values stored in the buffer memory are cleared when the station in which the Ethernet module is installed is powered on or reset. (They are not cleared during the initial processing.)
Normally, it is not necessary to read this area; so, read it as needed during maintenance.

(6) HTTP status storage area (address: 5101H to 5177H)

(a) Error log pointer (address: 5101H)

- This address stores the error log block number where the latest error logging is registered.
  0 : No error. (No registration of error log)
  1 or more : Error log block number where the latest error logging is registered
  ✻ If the pointer value is "16", it means that the latest error logging is registered in the error log block 16 area.
- When 17 or more receiving errors occur, the registration of error logging starts from error log block 1 area again.

POINT
The error log block area consists of 16 error log blocks that have the same data order.

(b) Log counter (HTTP response codes 100 to 199) (address: 5102H)
Log counter (HTTP response codes 200 to 299) (address: 5103H)
Log counter (HTTP response codes 300 to 399) (address: 5104H)
Log counter (HTTP response codes 400 to 499) (address: 5105H)
Log counter (HTTP response codes 500 to 599) (address: 5106H)
The log counters store how many times the Ethernet module has sent the HTTP response codes to the Web browser.

(c) Error log block : HTTP response code (address: starting from 5108H)
This error log block stores the HTTP response code when an error occurs. (See Section 11.3.)

(d) Error log block : Communication counterpart IP address (address: 5109H, 510AH and up)
This error log block stores the IP address of the server when an error occurs.

(e) Error log block : Error occurrence time (address: starting from 510BH)
This block stores the time when the error occurs in BCD code.

b15 to b8	b7 to b0
Month (01H to 12H)	Lower 2-digitals of year (00H to 99H)
Hour (00H to 23H)	Date (01H to 31H)
Second (00H to 59H)	Minutes (00H to 59H)
Higher 2-digitals of year (00H to 99H)	Day of the week (0 to 6)

11 - 17

(7) E-mail status storage area (addresses: 5870H to 5FFFH)

* When the storage count exceeds FFFFH times, the count starts from 0H again.

(a) E-mail status storage area for reception (addresses: 5870H to 5B38H)

1) Number of mails remaining on the server (address: 5870H)
- This area stores the number of mails remaining when the Ethernet module inquires to the receiving mail server.
  - 0 : No received mail in the server
  - 1 to 15 : The number of mails remaining in the server
  - 16 The number of mails in the server is 16 or more

2) Dedicated instruction normal completion count (address: 5871H)
- This area stores a cumulative count of how many times the dedicated instruction (MRECV) completed normally.
  - 0 : The MRECV instruction is not executed or no executions have resulted in normal completion.
  - 1 or more : Cumulative count of normal completions of the MRECV instruction

3) Dedicated instruction abnormal completion count (address: 5872H)
- This area stores a cumulative count of how may times the dedicated instruction (MRECV) completed abnormally.
  - 0 : The MRECV instruction is not executed or no executions have resulted in abnormal completion.
  - 1 or more : Cumulative count of abnormal completions of the MRECV instruction

4) Normal receiving count (address: 5873H)
- This address stores a cumulative count when the Ethernet module transferred received mails to the mail buffer data area.
  - 0 : No mail is transferred.
  - 1 or more : The number of normal mail transfer completions

5) Attached file receiving count (address: 5874H)
- This address stores a cumulative count of how many times the Ethernet module received mails with files attached.
  - 0 : No receiving of mail with files attached
  - 1 or more : The number of normal completions of mail receiving with files attached

6) Server inquiry count (address: 5875H)
- This address stores a cumulative count of inquiries made to the receiving mail server according to the parameter settings. (See Chapter 2 of the User's Manual (Application).)
  - 0 : No inquiry is made to the server.
  - 1 or more : Cumulative count of inquiries to the server

7) Server communication error count (address: 5876H)
- This address stores a cumulative count of communication error occurrences that are returned when making inquiries to the receiving mail server.
  - 0 : No communication error between servers, or no inquiry has been made
  - 1 or more : Cumulative count of communication error occurrences

8) Error log write count (address: 5877H)
   - This address stores a cumulative count of registrations made to the receiving error log block area.
     0 : No error, or no inquiry has been made to the server.
     1 or more : Cumulative count of registrations made to the error log block area

9) Receiving error log write pointer (address: 5878H)
   - This address stores the error log block number where the latest error logging is registered.
     0 : No error. (No registration of error log)
     1 or more : Error log block number where the latest error logging is registered
     * If the pointer value is "16", it means that the latest error logging is registered in the error log block 16 area.
   - When 17 or more receiving errors occur, the registration of error logging starts from error log block 1 area again.

POINT
The error log block area consists of 16 error log blocks that have the same data order.

10) Error log block – Error code (address: starting from 5879H)
    - This block stores the error codes that indicate the contents of errors. (See Section 11.3.)

11) Error log block – Command code (address: starting from 587AH)
    - This block stores the system command codes for the error causing messages.

12) Error log block – From (address: starting from 587BH)
    - This block stores eight words from the beginning of the sending source mail address of an error causing e-mail during communication with the mail server, in ASCII code characters.
      (Example)
      If the sending source mail address was "use@from.add.sample.co.jp",
      "use@from.add.sam" is stored as ASCII code characters.

13) Error log block – Date (address: starting from 5883H)
    - This block stores the time and date on which the e-mail is received in BCD code.

b15 to b8	b7 to b0
Month (01H to 12H)	Lower 2-digits of year (00H to 99H)
Hour (00H to 23H)	Date (01H to 31H)
Second (00H to 59H)	Minute (00H to 59H)
Higher 2-digits of year (00 to 99)	Day of the week (0 to 6)

14) Error log block – Subject (address: starting from 5887H)
    - This block stores 30 words from the beginning of the Subject of the e-mail.
    - A Subject is not stored successfully if it contains characters other than alphanumeric and ASCII code.

11 - 19

(b) E-mail status storage area for sending (addresses: 5B39H to 5CA0H)

1) Dedicated instruction normal completion count (address: 5B39H)
- This area stores a cumulative count of how many times the dedicated instruction (MSEND) completed normally.
  - 0 : The MSEND instruction is not executed or no executions have resulted in normal completion.
  - 1 or more : Cumulative count of normal completions of the MSEND instruction

2) Dedicated instruction abnormal completion count (address: 5B3AH)
- This area stores a cumulative count of how many times the dedicated instruction (MSEND) completed abnormally.
  - 0 : The MSEND instruction is not executed or no executions have resulted in abnormal completion.
  - 1 or more : Cumulative count of abnormal completions of the MSEND instruction

3) Number of mails normally completed (address: 5B3BH)
- This address stores a cumulative count of how many times the Ethernet module transferred send mails to the Send mail server.
  - 0 : No mail is sent.
  - 1 or more : The number of normal completions of mail sending

4) Attached file sending count (address: 5B3CH)
- This address stores a cumulative count of how many times the Ethernet module sent mails with files attached.
  - 0 : No sending of mail with files attached.
  - 1 or more : The number of normal completions of mail sending with files attached

5) Sending to the server count (address: 5B3DH)
- This address stores a cumulative count of sending to the send mail server.
  - 0 : No mail is sent to the server.
  - 1 or more : Cumulative count of sending to the server

6) Number of mails abnormally completed (address: 5B3EH)
- This address stores a cumulative count of communication error occurrences that are returned when requesting sending to the transmitting mail server.
  - 0 : No communication error between servers, or no transmission has been made
  - 1 or more : Cumulative count of communication error occurrences

7) Error log write count (address: 5B3FH)
- This address stores a cumulative count of registrations made to the sending error log block area.
  - 0 : No error, or no inquiry has been made to the server.
  - 1 or more : Cumulative count of registrations made to the error log block area

8) Sending error log write pointer (address: 5B40H)
- This address stores the error log block area number where the latest sending error logging is registered.
  0 : No error. (No registration of sending error log)
  1 or more : Error log block number where the latest sending error logging is registered
  * If the pointer value is "8", it means that the latest error logging is registered in the sending error log block 8 area.
- When 9 or more sending errors occur, the registration of sending error logging starts from sending error log block 1 area again.

POINT
The sending error log block area consists of eight error log blocks that have the same order of data items.

9) Error log block – Error code (address: starting from 5B41H)
- This block stores the error codes that indicate the contents of errors. (See Section 11.3.)

10) Error log block – Command code (address: starting from 5B42H)
- This block stores the system command codes for the error causing messages.

11) Error log block – To (address: starting from 5B43H)
- This block stores eight words from the beginning of the sending source mail address of an error causing e-mail during communication with the mail server, in ASCII code characters.

(Example)
If the sending source mail address was "use@from.add.sample.co.jp", "use@from.add.sam" is stored as ASCII code characters.

12) Error log block – Date (address: starting from 5B4BH)
- This block stores the time and date on which the e-mail is sent in BCD code.
- The order of date and time to be stored is the same as for the date and time of e-mail reception shown in (a) 13).

13) Error log block – Subject (address: starting from 5B4FH)
- This block stores 15 words from the beginning of the Subject of the e-mail.

11.3.1 End codes (Complete codes) returned to an external device during data communication

This section explains the end codes (complete codes) that are added to responses when communicating using the fixed buffer, the random access buffer or the MC protocol.
For more details on the error codes that are added to responses during communication using A compatible 1E frames through the MC protocol, see Section 11.3.2.
For more details on the end codes (error codes) that are stored in the buffer memory of the Ethernet module, see Section 11.3.3.

<table>
<tr><th rowspan="2">End code</th><th rowspan="2">Description</th><th rowspan="2">Processing</th><th colspan="3">Communication function</th></tr>
<tr><th>Fixed</th><th>Random</th><th>MC</th></tr>
<tr><td>00H</td><td>• Normal completion</td><td>—</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>02H</td><td>• Designation of device range of devices to be read/written from/to is incorrect.</td><td>• Check and correct the designated head device and number of points.</td><td></td><td></td><td>○</td></tr>
<tr><td>50H</td><td>• Codes for command/response type of subheader are not within the specifications.
<table><tr><th>Communication processing</th><th>Command/ response type</th></tr><tr><td>Communication using fixed buffer</td><td>60H</td></tr><tr><td>Communication using random access buffer</td><td>61H, 62H</td></tr><tr><td>Communication using the MC protocol</td><td>00H to 3CH</td></tr></table>• In communication using the fixed buffer, if the data length setting is less than the actual data count, the remaining data is determined as the second data and processed. In this case, a subheader undefined command type error may occur.</td><td>• Check and correct command/response type set by an external device. (The Ethernet module automatically adds command/response type; the user does not need to set these.) See REMARK in Section 11.3.3.
• Check and correct the data length.</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>51H</td><td>• In communication using the random access buffer, the head address designated by an external device is set outside the range from 0 to 6143.</td><td>• Check and correct the designated head address.</td><td></td><td>○</td><td></td></tr>
<tr><td>52H</td><td>• In communication using the random access buffer, head address + data word count (depending on the setting when reading) designated by an external device exceeds the range from 0 to 6143.
• Data for the designated word count (text) cannot be sent in one frame. (The value of the data length and amount of text communicated are not in the allowable range.)</td><td>• Check and correct the head address and data word count
• Correct the number of points read/written.</td><td></td><td>○</td><td></td></tr>
<tr><td>54H</td><td>• When “ASCII code communication” is selected in [Operational settings] – [Communication data code] with GX Developer, ASCII code data that cannot be converted to binary code was received from an external device.</td><td>• Check and correct the send data of the external device.</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>55H</td><td>• When [Operational setting] – [Enable Write at RUN time] is set to disable (no check mark) with GX Developer, an external device requests to write data while the PLC CPU is in the RUN status.
• An external device requests writing a parameter, sequence program or microcomputer program while the PLC CPU is in the RUN status. (Not related to the settings in [Operational settings] – [Enable Write at RUN time with GX Developer])</td><td>• Write data by setting [Enable Write at RUN time] to enable (with check mark). However, writing a parameter, sequence program, or microcomputer program is not allowed while the CPU is in the RUN status.
• Write such data by placing the PLC CPU in the STOP status.</td><td></td><td></td><td>○</td></tr>
</table>

Fixed : Fixed buffer communication Random : Random access buffer communication MC : Communication using the MC protocol

11 - 22

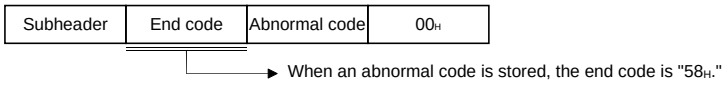
End code	Description	Processing	Communication function		
			Fixed	Random	MC
56H	• Device designation from an external side is incorrect.	• Correct the device designated.			○
57H	• The number of points for a command designated by an external device exceeds the maximum number of processing points (number of processing that can be executed per communication) for each processing. • Addresses from the head address (head device number and head step number) to the designated points exceed the maximum addresses (device number and step number).	• Correct the designated points or the head address (device number and step number).			○
	• Byte length of a command does not conform to the specifications. • When writing data, the set number of data points written is different from the value of the designated number.	• Check the data length of the command and redo the data setting.			○
	• Monitoring was requested even though monitor data is not registered.	• Register the monitor data.			○
	• When reading/writing in a microcomputer program, an address after the end of parameter setting range is designated.	• Addresses after the last address cannot be read/written from/to. Correct the address designated.			○
	• In the block number designation of the extension file register, a block number exceeding the range of corresponding memory cassette size is designated.	• Correct the block number.			○
58H	• A head address (head device number and head step number) of a command designated by an external device is set outside the range that can be designated. • When reading from/writing to a microcomputer program or file register (R), values exceeding the PLC CPU's parameter setting range was designated.	• Designate the appropriate values within the range that are allowed for each processing.			○
	• A block number designated for an extension file register does not exist.	• Correct the block number.			○
	• Cannot designate a file register (R).	• Check the device.			○
	• A word device is designated for a command for bit devices. • The head number of bit devices is designated by a value other than a multiple of 16 in a command for word devices.	• Correct the command or the designated device.			○
59H	• Cannot designate an extension file register.	• Check the device.			○
5BH	• The PLC CPU and the Ethernet module cannot communicate. • The PLC CPU cannot process requests from an external device.	• Fix the faulty parts by referring to the abnormal codes appended to the end codes (see Section 11.3.2).			○
60H	• Communication time between the Ethernet module and the PLC CPU exceeded CPU monitoring timer value.	• Increase the CPU monitoring timer value.			○
63H	• In fixed buffer communication, the remote password status of the port for the destination Ethernet module is in the lock status.	• Perform fixed buffer communication after unlocking the remote password using the MC protocol. • Do not set the port for fixed buffer communication as a target for the remote password check.	○	○	
A0H to FFFFH	• The content of the error and the error handling for each end code are the same as for the error codes from A0H to FFFFH that are stored in the buffer memory. Take a necessary action by referring to the description of the applicable error code shown in Section 11.3.3.				

11 - 23

11.3.2 Abnormal codes returned during communication using A compatible 1E frames

This section explains the abnormal codes (error codes) that are added to responses when communicating using the MC protocol and A compatible 1E frames. (An abnormal code is added only when an end code is "5B".)
For more details on the end codes (error codes) that re added to responses, see Section 11.3.1.
For more details on the end codes (error codes) that are stored in the buffer memory of the Ethernet module, see Section 11.3.3.

Response format



Error code (hexadecimal)	Error	Description of error	Corrective action
10H	PC number error (PLC number error)	A station with the specified PC number does not exist. (1) The PC number designated with a command is neither "FF" of the local station nor any of the station numbers designated with the MELSECNET link parameters.	(1) Change the PC number to "FF" of the local station or a station number set in the link parameter, and communicate again.
11H	Mode error	Poor communication between the Ethernet module and the PLC CPU (1) After the Ethernet module receives a request successfully from an external device, the Ethernet module and the PLC CPU could not communicate for some reason (noise, etc.).	(1) Communicate again. If an error occurs again, check noise, etc. and replace the Ethernet module, then communicate again.
12H	Intelligent function module designation error	Intelligent function module error (1) The intelligent function module with the buffer memory that can be communicated with, does not exist at the location designated by the intelligent function module number. (For example, the corresponding place is for an input/output module or an empty slot.)	(1) Change the data content designated in the control procedure or change the installation location of the intelligent function module, and communicate again.
18H	Remote error	Remote RUN/STOP not accessible. Another module (other Ethernet module, etc.) has already executed remote STOP/PAUSE.	(1) Check whether or not other module has executed remote STOP/PAUSE. If one of these commands has been executed, cancel it and communicate again.
1FH	Device error	Invalid device specification	(1) Review the specified device. (2) Does not access any non-existent device.
20H	Link error	The CPU module of the request destination is disconnected from the data link.	Check whether or not the PLC CPU with the station number set as PC number is disconnected. Remove the cause of disconnection and connect again.
21H	Intelligent function module bus error	Cannot access the memory of the intelligent function module. (1) The control bus to the intelligent function module is faulty. (2) The intelligent function module is faulty.	One of the following hardware is faulty: the PLC CPU, base unit, intelligent function module, or Ethernet module. Please consult your nearest dealer.

11 - 24

11.3.3 Error codes stored in the buffer memory

When an error occurs at each data communication processing between the Ethernet module and an external device, the error code (abnormal code) is stored in the buffer memory of the Ethernet module. This section explains the contents of this type of errors and error handling procedures.
The "Storage destination" column in the error code list indicates the buffer memory where the applicable error code is stored.
The names used for the explanation indicated in the "Storage destination" column correspond to the buffer memory error code storage areas shown in the table below. (Error codes whose storage destination is not written are returned to the external device.)
Note that the buffer memory may store error codes of the messages returned from the external device. For error codes not shown in this manual, refer to the manual of the external device and check the returned messages.

Name used for explanation	Buffer memory	Buffer memory address
Initial	Initial abnormal code area	69H (105)
Open	Open abnormal code area	7CH (124)…
Fixed sending	Fixed buffer sending abnormal code area	7DH (125)…
Connection	Connection end code/error log area	7EH (126)…
Error code	Error code/end code area	E5H (229)…
Data link	((S1) + 1 for data link instructions)	—
HTTP log	HTTP response code area	5108H (20744)
E-mail log	E-mail error log area	5879H (22649)…
Dedicated instruction	(Complete status area for dedicated instructions)	—

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
02H	The content of these errors and the error handling for each error code is the same as the end codes (02H to 0063H) returned to the external device. Check the explanation of the applicable code shown in Section 11.3.1, and take actions accordingly.						○	○			
0050H						○	○				
0051H							○				
0052H					○	○	○				
0054H						○	○				
0055H						○	○				
0056H						○	○				
0057H						○	○				
0058H						○	○				
0059H						○					
005BH	Read and handle the error code and end code area.					○	○				
0060H						○	○				
0063H					○						
00A0H	Requests that cannot be designated for applicable connection.	• Review the content of request. • Correct the open settings. (See Section 5.5)				○					
00A1H	Contents of request cannot be analyzed because the length of text area or request data length is too short.	• Review and correct the length of the text area or the request data length of the Qn header, and send to the Ethernet module again.				○					
00A2H	Requests that cannot be processed.	• Correct the content of request and command.					○				
3E8H to 4FFFH	(Errors detected by the PLC CPU)	• Handle by referring to the troubleshooting section of the CPU User's Manual (Hardware Design, Maintenance and Inspection).				○	○				

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial	Open	Fixed sending	Connection	Error code	Data link	HTTP log	E-mail log	Dedicated instruction
			69H	7CH	7DH	7EH	E5H	—	5108H	5879H	—
7000H to 7FFFH	(Errors detected by serial communication module, etc.)	• Take corrective action, referring to the Serial Communication Module User's Manual, etc.					○				
B000H to BFFFH	(Errors detected by CC-Link module)	• Take corrective action, referring to the CC-Link System Master/Local Module User's Manual.					○				
C001H	• At initial processing, the IP address setting value of the Ethernet module is incorrect. • When using the router relay function, the setting value of the sub-net mask field is incorrect.	• Correct the IP address. Set the class to A/B/C. • Correct the sub-net mask.	○				○				
C002H	At initial setting, some of the various timer setting values are outside the allowable range.	• Review and correct the necessary timer values at initial processing.	○				○				
C003H	At initial setting, the setting value of the automatic open UDP port number is outside the allowable range.	• Correct the automatic open UDP port number.	○				○				
C004H	The setting value of the sub-net mask field is incorrect.	• Correct the sub-net mask and execute the initial processing again.	○				○				
C005H	• The setting value of the default router IP address for the router relay function is incorrect. • Network address (network address after sub-net mask) of the default router IP address is different from the network address of the local station's Ethernet module IP address.	• Correct the default router IP address and execute the initial processing again. • Set the network address to the same network address as the local station's Ethernet module IP address.	○				○				
C006H	The setting value of the sub-net address for the router relay function is incorrect.	• Correct the sub-net address and execute the initial processing again.	○				○				
C007H	• The setting value of the router IP address for the router relay function is incorrect. • Network address (network address after sub-net mask) of router IP address is different from the network address of the local station's Ethernet module IP address.	• Correct the router IP address and execute the initial processing again. • Set the network address to the same network ID as the local station's Ethernet module IP address.	○				○				
C010H	At open processing, the setting value of the Ethernet module port number is incorrect.	• Correct the port number.	○	○			○				
C011H	At open processing, the setting value of an external device's port number is incorrect.	• Correct the port number.		○	○		○				
C012H	The port number set is used in a connection already opened by TCP/IP.	• Review and correct the port numbers of the Ethernet module and external device.		○			○				
C013H	The port number used in a connection already opened is set in UDP/IP open processing.	• Review and correct the port number of the Ethernet module.		○			○				
C014H	Initial processing and open processing of the Ethernet module is not completed.	• Execute the initial processing and open processing.			○		○				

11 - 26

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial	Open	Fixed sending	Connection	Error code	Data link	HTTP log	E-mail log	Dedicated instruction
			69H	7CH	7DH	7EH	E5H	—	5108H	5879H	—
C015H	At open processing, the setting value of an external device's IP address is incorrect.	• Correct the IP address. Set the class to A/B/C.		○	○		○				
C016H	The pairing open processing of the connection (or the next connection) designated for pairing open has already been completed.	• Check that the open processing of both of the target connections for pairing open is not executed. • Review the combination for pairing open.		○			○				
C017H	A connection could not be established during the open processing of a TCP connection.	• Check the operation of the external device. • Check the open processing of the external device. • Correct the open settings of the communication parameters. • Review the Ethernet module's port number and the IP address/port number and open system of the external device. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty.		○			○				
C018H	The setting value of an external device's IP address is incorrect. * When TCP is used, FFFFFFFFH cannot be set as an IP address.	• Correct the IP address.		○			○				
C020H	Data length exceeds the allowable range.	• Correct the data length. • If the data transmitted is larger than the allowable size, divide and then send it.			○		○				
C021H	An abnormal end response was received after a transmission using fixed buffers.	• Read the end code of the response from the connection end code/error log area, and handle as needed.			○		○				
C022H	• A response could not be received within the response monitoring timer value. • The applicable connection was closed during waiting for a response.	• Check the operation of the external device. • Review and correct the response monitoring timer value. • Check the open status of the applicable connection.			○		○				
C023H	• The open processing for the applicable connection is not completed. • The applicable connection is closed.	• Execute the open processing of the applicable connection.			○		○				
C030H	A sending error occurred.	• Check the operation of the transceiver and external device. * Use a transceiver that allows the SQE test. • Send after an arbitrarily selected time has elapsed because packets may be congested on the line. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty. • Conduct a self-diagnosis test to see whether or not the Ethernet module is faulty.		○	○		○				
C031H	A sending error occurred.										

11 - 27

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C032H	A TCP ULP timeout error occurred in the TCP/IP communication. (An external device does not return ACK.)	• Check the operation of the external device. • Correct the TCP ULP timeout value and execute the initial processing again. • Send after an arbitrarily selected time has elapsed because packets may be congested on the line. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty.		○	○		○				
C033H	An external device side with the set IP address does not exist.	• Review and correct the external device's IP address and the Ethernet address. • If the external device has the ARP function, set the default value. If not, set the Ethernet address of the external device. • Check the operation of the external device. • Send after an arbitrarily selected time has elapsed because packets may be congested on the line. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty.		○	○		○				
C035H	The existence of an external device could not be confirmed within the response monitoring timer value.	• Check the operation of the external device. • Review and correct each setting value for the existence confirmation. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty.		○	○		○				
C036H	Send processing cannot be executed because a cable is not yet connected or disconnected.	• Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty. • Conduct a loopback test to see that the line is not faulty. • Conduct a self-diagnosis test to see that the Ethernet module is not faulty.					○				
C040H	• Not all the data could be received within the response monitoring timer value. • Sufficient data for the data length could not be received. • The remaining part of a message divided at the TCP/IP level could not be received within the response monitoring timer value.	• Review and correct the data length of the communication data. • Review and correct each setting value at the initial processing because the packets may be congested on the line. • Send the same data from the external device again.		○	○		○				
C041H	When TCP is used, the checksum of the receive data is incorrect.	• Review the checksum on the external device side and send the correct value. • Investigate the conditions of the line (noise, distance between the line and power line, contact of each device, etc.)			○		○				
C042H	When UDP is used, the checksum of the receive data is incorrect.				○		○				
C043H	The checksum in header of IP packet received is incorrect.				○		○				

11 - 28

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C044H to C048H	An error packet of ICMP was received.	• Check the operation of the external device. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty.			○		○				
C049H	An error packet of ICMP was received.	• Check the operation of the external device. • Send after an arbitrarily selected time has elapsed because packets may be congested on the line. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty. • Correct the IP assembly timer value of the external device existence timer timeout.			○		○				
C04AH	An error packet of ICMP was received. (An IP assembly timeout error occurred in an external device.)				○		○				
C04BH	An IP assembly timeout error occurred. (The remaining part of divided data could not be received and a timeout occurred.)	• Check the operation of the external device. • Send after an arbitrarily selected time has elapsed because packets may be congested on the line. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty. • Correct the IP assembly timer value and execute the initial processing again.			○		○				
C04CH	Cannot send because there is no space in the internal buffer, e.g. the IP header buffer.	• Send the same data again and check that the response is received.			○		○				
C04DH	• In a message the Ethernet module received through automatic open UDP port communication or non procedure fixed buffer communication, the data length designated in the application data field is incorrect. • Not all the receive data can be stored.	• Review the data length. • Review the text size length so that the text data becomes less than the receive buffer memory size.				○	○				
C050H	ASCII code data that cannot be converted to binary code is received when ASCII code communication is set in the operational settings of the Ethernet module.	• Select binary code communication in the operational settings, and restart the Ethernet module. • Correct the data sent from the external side and send again.			○		○				
C051H to C054H	The number of read/write points is outside the allowable range.	• Correct the number of read/write points and send to the Ethernet module again.			○		○				
C055H	The number of file data read/write points is outside the allowable range.	• Correct the number of read/write points (or byte points) and send to the Ethernet module again.			○		○				
C056H	• Read/write request exceeds the maximum address. • Address is 0.	• Correct the head address or the number of read/write points and send to the Ethernet module again. (The maximum address must not be exceeded.)			○		○				

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C057H	Request data length does not match with data count in character area (part of text).	• Review and correct the content of the text area or the request data length of Qn header, and send to the Ethernet module again.			○		○				
C058H	Request data length after ASCII-binary conversion does not match with data count in character area (part of text).	• Review and correct the content of the text area or the request data length of Qn header, and send to the Ethernet module again.			○		○				
C059H	Incorrect designation of command and subcommand.	• Review the content of the request.			○		○				
C05AH	The Ethernet module cannot read from/write to the designated device.	• Examine the device to be read/written.			○		○				
C05BH					○		○				
C05CH	The content of a request is incorrect. (Requesting read/write in bit units to word devices.)	• Correct the content of the request and send to the Ethernet module again. (Correction of subcommand, etc.)			○		○				
C05DH	Monitor registration is not performed.	• Monitor after performing monitor registration.			○		○				
C05EH	Communication time between the Ethernet module and PLC CPU exceeded the CPU monitoring timer.	• Increase the CPU monitoring timer value. • Check whether or not the PLC CPU operates normally.			○		○				
C05FH	The request could not be executed on the target PLC.	• Correct the network number and PC number. • Correct the content of the read/write request.			○		○				
C060H	The content of a request is incorrect. (Incorrect data was designated for bit devices, etc.)	• Correct the content of the request and send to the Ethernet module again. (Correction of data, etc.)			○		○				
C061H	Request data length does not match with the number of data in character area (part of text).	• Review and correct the content of the text area or the request data length of the Qn header, and send it to the Ethernet module again.			○		○				
C062H	When online correction was inhibited, write operation via MC protocol (QnA-compatible 3E frame) was performed for the remote I/O station.	• When performing write operation to the remote I/O station via the MC protocol (QnA-compatible 3E frame), "enable" the online correction setting of the operation setting.					○				
C070H	Extension of device memory cannot be designated for the target station.	• Read/write without designating extension. ＊ Device memory extension can be designated only for stations with Ethernet modules installed and Q/QnACPUs that connect via the MELSECNET/H, MELSECNET/10.			○		○				
C071H	Too many devices points to read/write for other than the Q/QnACPU.	• Correct the number of device points to read/write and send it to the Ethernet module again.			○		○				
C072H	The content of a request is incorrect. (Requested read/write in bit units to word devices.)	• Check whether the content can be requested to the target PLC CPU. • Correct the content of the request and send it to the Ethernet module again. (Correction of subcommand, etc.)			○		○				
C073H	Request was not supported by the Ethernet module of the target PLC CPU. (Double word address points was designated for other than Q/QnACPU, etc.)	• Review the content of the request.			○		○				

11 - 30

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C074H	Request could not be executed on the target PLC.	• Correct the network number and PC number. • Correct the content of the read/write request.			○		○				
C080H	The IP address of an external device could not be obtained during MELSECNET/H, MELSECNET/10 relay communication or data link instruction communication.	• Set the Station No. <-> IP information for the Ethernet module. • Change the parameter conversion method for the MELSECNET/H, MELSECNET/10.					○	○			
C081H	The end processing for the Ethernet module has been completed and data link instruction communication cannot be checked.	• Execute the end processing of the Ethernet module after all communication is over.					○	○			
C082H	Communication processing was abnormally completed in following communication. • Communication with GX Developer (UDP/IP) • MELSECNET/H, MELSECNET/10 relay.	• Check that the relay station/external station operates normally. (If communication is continuing, it is not necessary to take actions for the error.) • Check whether or not the cable connection between the local station and external station is faulty. • If some load is applied to the line, reduce the load.					○				
C083H	Communication processing was abnormally completed in data link instruction communication.						○	○			
C084H	Communication processing was abnormally completed in data link instruction communication	• Check that the local station/relay station and external station operate normally. • Check whether or not the cable connection between the local station and external station is faulty. • Increase TCP resend timer value.					○				
C085H	The local station's channel designated by another station in data link instruction communication is currently in use.	• Execute the request from the other station again.					○	○			
C086H	A message exceeding the receive message size was received.	• Correct the send message size from the requesting source.					○				
C087H	There is an error in the IP address for the Station No. <-> IP information setting for MELSECNET/H, MELSECNET/10 relay communication.	• In the Station No. <-> IP information setting, set the IP address of the target device for MELSECNET/H, MELSECNET/10 relay communication.					○				○
C0B2H	Insufficient free space in receive buffer of the relay station for MELSOFT connection or data link instruction, or the communication requested station. (Receive buffer full error)	• Provide sufficient time between request intervals • Reduce the number of request nodes. • Wait for a response to the request and issue the next request after receiving the response. • Review the time-out value.					○	○			
C0B3H	A request that could not be processed from the PLC CPU was made.	• Review the content of the request. • Correct the network number and PC number.					○	○			
C0B5H	Data that could not be processed by the PLC CPU/Ethernet module was designated.	• Review the content of the request. • Cancel the current request.					○	○			
C0B6H	A channel number is outside the allowable range.	• Designate the channel number in the range from 1 to 8.					○	○			
C0B7H	A channel number currently in use was designated.	• Change the channel number. • Execute after the current communication is completed.					○	○			

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C0B8H	Network number and PC number are outside the allowable range. A response from the PLC CPU is faulty.	• Correct the network number and PC number. • Check the operation of the PLC CPU.					○				
C0B9H	The open processing of the applicable connection is has not been completed.	• Execute the open processing. • Check the operation of the external device.			○		○				
C0BAH	Cannot acknowledge sending request since the close processing is being executed via the CLOSE instruction.	• Execute the open processing and make a sending request.			○		○				
C0BBH	System error • The OS detected any error.	(＊1)									
C0BCH	Designated communication line is closed.	• Open the communication line. • Review the target connection number.			○	○	○				
C0BDH	Cannot send by acknowledging continuous requests.	• Check whether or not requests are made continuously without waiting for responses.			○	○	○				
C0BEH C0BFH	System error • The OS detected any error.	(＊1)									
C0C0H	The receiving completion confirmation signal was turned on when the receiving completion signal was not on.	• Review and correct the program.					○				
C0C1H	The transmission interval of UDP is too short.	• Check whether or not sending requests are repeated. • Make the sending interval longer.					○				
C0C2H to C0C4H	System error • The OS detected any error.	(＊1)									
C0C5H	• A sending request was made to an external device whose class/network address are different from those of the local station when the router relay function is not used. • The setting of the router relay parameter is incorrect.	• Execute the initial processing by setting that the router relay function should be used. • Set the correct data for the router relay parameter. • Correct the IP address of the external device and execute the open processing. • Check that the network address is correct. When it is changed, execute the initial processing again.		○	○		○				
C0C6H	System error • The OS detected any error.	(＊1)									
C0C7H	An Ethernet module system error occurred.	• Execute the initial processing again. • Execute the processing by referring Section 11.4 POINT (3).	○		○		○				
C0C8H to C0CAH	System error • The OS detected any error.	(＊1)									
C0CBH	Another sending request was made when the sending processing has not been completed.	• Make the next sending request after the previous sending is completed.			○		○				
C0CCH C0CFH	System error • The OS detected any error.	(＊1)									
C0D0H	Incorrect data length was designated.	• Review the designated value of the data length.					○	○			
C0D1H	Incorrect resent count was designated.	• Review the designation of the resent count.					○	○			

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C0D2H	Incorrect watchdog time was designated.	• Review the designated value of the watchdog timer.					○	○			
C0D3H	The number of relay stations in MELSECNET/H, MELSECNET/10 exceeded the allowable count.	• Check the designated value of the destination. • Review the Station No. <-> IP information between the local station and destination. • Review the system specifications.					○				
C0D4H							○				
C0D5H	Incorrect retry count was designated.	• Review the retry count.					○	○			
C0D6H	The designations of network number or station number were incorrect.	• Review the content of the destination designation. • Review the designated values of the destination.					○	○			
C0D7H	Parameter values set by GX Developer have not been loaded.	• Set parameters using GX Developer and load them to the CPU module.					○	○			
C0D8H	The number of blocks exceeded the range.	• Correct the designation value for the number of blocks.					○				
C0D9H	Incorrect subcommand value was designated.	• Correct the designated value for the subcommand.					○				
C0DAH	A response to the PING test could not be received within the time of the communication time check.	• Review the IP address/host name of the Ethernet module for the PING test. • Change the status of the Ethernet module for the PING to allow communication (to the status after the initial processing is completed).									○
C0DBH	There is an error in the IP address/host name of the Ethernet module for the PING test.	• Review the IP address/host name of the Ethernet module for the PING test.									○
C0DCH	System error • The OS detected any error.	(＊1)									
C0DDH											
C0DEH	Data could not be received within the designated watchdog time.	• Review the designated value of the watchdog timer. • Review the designated value of the channel number. • Check the status of the sending source station and relay station.					○	○			
C0DFH	System error • The OS detected any error.	(＊1)									
C0E0H to C0EFH	An error was detected in the PLC CPU.	• Check if the PLC CPU and intelligent function modules are correctly mounted to the base or not. • Check that the PLC CPU does not remain in the reset status. • Check that no error has occurred in the PLC CPU. If an error is found, take corrective actions according to error description of the PLC CPU. • Replace the power supply module, PLC CPU and/or intelligent function module.					○				
C0F0H	An Ethernet module RAM abnormality was detected in the hardware test.	• Conduct a hardware test again. If an abnormality is detected again, the Ethernet module hardware may be faulty. Consult your nearest branch office or dealer with the details of the errors.					○				
C0F1H	An Ethernet module ROM abnormality was detected in the hardware test.	• Conduct a hardware test again. If an abnormality is detected again, the Ethernet module hardware may be faulty. Consult your nearest branch office or dealer with the details of the errors.					○				

11 - 33

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C0F3H	A module system error (serious abnormality) was detected in the CPU.	• Correct the causes of error in the local station CPU.					○				
C0F4H to C0F6H	System error • The OS detected any error.	(＊1)									
C0F7H	An error occurs in the self refrain test	• Send after an arbitrarily selected time has elapsed because packets may be congested on the line. • Check that the connection cable is not dislocated. • Check that connections to the transceiver and terminator are not faulty. • When other than above, the Ethernet module hardware may be faulty. Consult your nearest dealer with the details of the errors.					○				
C100H	System error • The OS detected any error.	(＊1)									
C101H	A response could not be received from the DNS client.	• Check the address of the DNS server. • Check whether or not it is possible to communicate with the DNS server using the Ping command. • Check that the IP addresses of the local station and DNS server are in the same class. (If the class is different, check the router setting.)								○	
C102H	A response from the SMTP layer could not be received.	• Check that the SMTP server name is registered in DNS. • Delete the SMTP server name, change to the IP address setting, and check the operation. • Check whether or not it is possible to communicate with the SMTP server using the Ping command.								○	
C103H	DNS settings incorrect.	• Check the DNS mail address. • Check the content of the DNS setting.									
C104H to C106H C110H	System error • The OS detected any error.	(＊1)									
C111H	A response could not be received from the DNS client.	• Check cable, hub, etc. • Check whether or not it is possible to communicate with the DNS server using the Ping command.								○	
C112H	A response from the POP3 layer could not be received.	• Check that the POP3 server name is registered in DNS. • Delete the POP3 server name, change to the IP address setting, and check the operation. • Check whether or not it is possible to communicate with the POP3 server using the Ping command.								○	

11 - 34

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C113H	An e-mail was received that did not have an attached file. (It will generate when the attached file is not read normally.)	• Designate an attached file on the sending side. • Check the program on the sending side. • When the sending source is the mail server, sending with the previous MSEND instruction failed. Check the destination of the MSEND instruction, etc. • Check that the sending side has the same e-mail specifications as the Ethernet module. (encode/decode, file format, etc.) • A server with unknown destination was received from the SMTP server.								○	
C114H	An e-mail was received whose attached file name was invalid.	• Check on the sending side whether the extension of the attached file is “bin” or "asc". • Check whether or not the mail is compressed or encrypted. • Check the destination of the MSEND instruction, etc. • A server with unknown destination was received from the SMTP server.								○	
C115H to C118H	System error • The OS detected any error.	(＊1)									
C119H	There is no received mail.	• Check that the sending side has the same e-mail specifications as the Ethernet module. (encode/decode, file format, etc.) • Check the mail information storage area of the buffer memory (address: 2682H), then read any received mail that is in the server.									○
C11AH	Failed to convert an e-mail to be received.	• Check whether or not the mail is compressed or encrypted. • Check that the sending side has the same e-mail specifications as the Ethernet module. (encode/decode, file format, etc.) • Check whether or not the file was divided on the sending side.								○	
C11BH	An e-mail was sent and an error mail was received from the mail server of the destination.	• A server with unknown destination was received from the SMTP server. (Received mail is stored in the mail buffer.) • Check that the part before "@" is correct in the mail address setting of the parameter settings. • Check that the part before "@" is registered to the destination mail server.								○	○
C11CH	Mail address not found.	• Check whether the mail setting of the parameter setting is correct or not. • When the mail server and Ethernet module are connected via the router, check whether the router setting is correct or not. • Test-send a mail to the address where it will be received without fail. When the test is normally completed, recheck whether the domain name after "@" is correct or not.								○	○
C11DH	The size of the attached file exceeded the allowable size.	• Check that the attached file is less than 6 k word. • Check that the sending side did not divide the file.								○	

11 - 35

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C120H	Could not open the SMTP server.	• Check that the port number of the SMTP server = 25. • Check whether or not it is possible to communicate with the SMTP server using the Ping command.								○	
C121H	Cannot communicate with the SMTP server. (Error response)	• Check if the SMTP server is busy.								○	
C122H	Cannot communicate with the SMTP server. (Abort)	• Check if the SMTP server is busy.								○	
C123H	Cannot communicate with the SMTP server. (Reset response)	• Check if the SMTP server is busy.								○	
C124H	A response from the SMTP server timed out.	• Check whether or not the SMTP server is faulty. • Check whether or not there is too much load on the network.								○	
C125H	Forcefully disconnected from the SMTP server.	• Check whether or not the SMTP server is faulty.								○	
C126H	Could not close the SMTP server.	• Check whether or not the SMTP server is faulty. • Check whether or not there is too much load on the network.								○	
C127H	Closing the SMTP server gave an error response.	• Check whether or not the SMTP server is faulty.								○	
C130H	Communication channel is closed because the service is not available.	• Check the status of the SMTP server.								○	
C131H	The SMTP server was performing processing and an error response was received.	• Check if a user name not registered in the server was designated. • Send again after arbitrary set time has passed.								○	
C132H	The SMTP server was performing processing and an error response was received. (Local error)	• Check the status of the SMTP server.								○	
C133H	The SMTP server was performing processing and an error response was received. (Insufficient memory area)	• Check the status of the SMTP server.								○	
C134H to C137H	System error • The OS detected any error.	(＊1)									
C138H	The SMTP server was performing processing and an error response was received. (Mailbox not found)	• Check that the Ethernet module's mail address is set correctly.								○	
C139H	System error • The OS detected any error.	(＊1)									
C13AH	The SMTP server was performing processing and an error response was received. (Exceeded the allocation of memory area)	• Check the status of the SMTP server.								○	
C13BH	The SMTP server was performing processing and an error response was received. (Illegal mail box name)	• Check that the Ethernet module's mail address is set correctly.								○	
C13CH	System error • The OS detected any error.	(＊1)									

11 - 36

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C140H	Could not open the POP3 server.	• Check that the port number of the POP3 server = 110. (For the Ethernet module, this is fixed to 110.) • Check whether or not it is possible to communicate with the POP3 server using the Ping command.								○	
C141H	Cannot communicate with the POP3 server. (Error response)	• Check if the POP3 server is busy.								○	
C142H	Cannot communicate with the POP3 server. (Abort)	• Check if the POP3 server is busy.								○	
C143H	Cannot communicate with the POP3 server. (Reset response)	• Check if the POP3 server is busy.								○	
C144H	Could not receive a response from the POP3 server.	• Check whether or not the POP3 server is faulty. • Check whether or not there is too much load on the network.								○	
C145H	Forcefully disconnected from the POP3 server.	• Check whether or not the POP3 server is faulty.								○	
C146H	Could not close the POP3 server.	• Check whether or not the POP3 server is faulty. • Check whether or not there is too much load on the network.								○	
C147H	Closing the POP3 server gave an error response.	• Check whether or not the POP3 server is faulty.								○	
C150H	POP3 server verification error.	• Check the status of the POP3 server.								○	
C151H	The Ethernet module's mail address (e-mail address setting parameter) is different from the account name in the mailbox on the server side.	• Check the account name in the mailbox on the server side and correct the mailbox account set for the Ethernet module.								○	
C152H	The Ethernet module's password (e-mail setting parameter) is different from the password on the server side.	• Check the password on the server side and correct the password set for the Ethernet module.								○	
C153H	An error occurred when getting the received mail list. (Failed to obtain the list of mail arrived at the POP3 server.)	• Reset the server inquiry time to the default value and reset the local station's PLC CPU.								○	
C154H	An error occurred when receiving a mail. (Cannot read e-mail from the POP3 server.)	• Check whether or not the mail is compressed or encrypted. • Check that the sending side has the same e-mail specifications as the Ethernet module. (encode/decode, file format, etc.)								○	
C160H	Received a response from the DNS server after timeout.	• Check whether or not there is too much load on the network. • Check the status of the DNS server.								○	
C161H	Could not receive a response from the DNS server.									○	

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C162H C163H C171H to C17FH	An error is returned from DNS server.	• Check if the DNS server's IP address setting is correct or not. • Check if the mail server name setting (SMTP server name, POP server name) is correct or not. • Check with the network administrator or similar person that the DNS function of the server set in the DNS setting" is being performed.								○	
C1A0H	An illegal request was made.	• Execute again. If the same error occurs, the Ethernet module's hardware may be faulty. Consult your nearest dealer with the details of the error.									○
C1A2H	A response to a request could not be received.	• Review and correct the response wait time.									○
C1A4H	A request or subrequest was incorrect.	• Correct the request or subrequest.									○
C1A5H	The designation of the target station or clear target were incorrect.	• Correct the designated value of the target station or clear target.									○
C1A6H	Incorrect connection number was designated.	• Designate the connection number within the range of 1 to 16.									○
C1A7H	Incorrect network number was designated.	• Correct the designated value of the network number.									○
C1A8H	Incorrect station number was designated.	• Correct the designated value of the station number.									○
C1A9H	Incorrect device number was designated.	• Correct the designated device value of the device number.									○
C1AAH	Incorrect device name was designated.	• Correct the designated value of the device number.									○
C1ACH	Incorrect resent count was designated.	• Correct the designated value of the resent count.									○
C1ADH	Incorrect data length was designated.	• Correct the designated value of the data length.									○
C1AEH	Incorrect mail sending/receiving data length and header length were designated.	• Correct the designated values of sending/receiving data length and header length. • Sending/receiving data length should be equal or longer than the header length.									○
C1AFH	Incorrect port number was designated.	• Correct the designated value of the port number.									○
C1B0H	The open processing of the designated connection is already completed.	• Execute the open processing after executing the close processing.									○
C1B1H	The open processing of the designated connection has not been completed.	• Execute the open processing.									○
C1B2H	The OPEN/CLOSE instruction is being executed in the designated connection.	• Execute after the OPEN/CLOSE instruction is completed.									○
C1B3H	Another sending/receiving instruction is being executed on the designated channel.	• Change the channel number. • Execute after the sending/receiving instruction is completed.									○
C1B4H	Incorrect arrival time was designated.	• Set the watchdog timer within the proper range.									○
C1B5H	Data could not be received within the designated watchdog time.	• Review the designated value of the watchdog timer.									○
C1B6H	Incorrect mail destination number was designated.	• Review the designated value of the mail designation number. • Review the sending mail address setting parameter.									○

Error code (abnormal code)	Description of error	Error handling	Storage destination								
			Initial 69H	Open 7CH	Fixed sending 7DH	Connection 7EH	Error code E5H	Data link —	HTTP log 5108H	E-mail log 5879H	Dedicated instruction —
C1B7H	A reading operation was executed while no receiving e-mail was stored in the mail buffer data area.	• Execute the MRECV instruction when the mail receiving flag of the mail information is "yes".									○
C1B8H	The RECV instruction was executed for the channel that had not received data.	• Review the execution condition of the RECV instruction. • Review the channel number.									○
C1B9H	The OPEN instruction cannot be executed for the specified connection.	• Review the connection number.									○
C1BAH	A dedicated instruction was executed while in the initialization uncompleted status	• Do not execute any dedicated instructions until after the initial processing is completed.									○
C1BBH	The target station CPU type is wrong.	• Review the value specified for the target station CPU type.									○
C200H	There is an error in the remote password.	• Review the remote password, then perform the remote password unlock/lock processing again.					○				
C201H	The remote password status of the port used for communication is in the lock status.	• Unlock the remote password, then perform communication.					○				○
C202H	When another station was accessed, the remote password could not be unlocked.	• When accessing another station, do not set a remote password in the relay station or station to be accessed, or set so they are not subject to the remote password check.					○				
C203H	System error • The OS detected any error.	(＊1)									
C204H	The device is different from the one requesting the remote password unlock.	• Request the remote password lock processing from the opposite device that has requested the remote password unlock processing.					○				
C205H	When another station was accessed, the remote password could not be unlocked.	• When accessing another station, do not set a remote password in the relay station or station to be accessed, or set so they are not subject to the remote password check.					○				
C206H	System error • The OS detected any error.	(＊1)									
C207H	Too many file name characters.	• Reduce the file name characters to within 255 characters.					○				
C300H	A response could not be received within the response monitoring timer value.	• Check the operation of the external device. • Review and correct the response monitoring timer value.					○				
F000H to FFFFH	(Errors detected by the MELSECNET/H, MELSECNET/10 network system)	• Refer to the MELSECNET/H, MELSECNET/10 Network System Reference Manual for details on error handling.					○				

＊1 Take corrective action in the following procedure.

1) Check whether the Ethernet module, power supply module and CPU module are mounted correctly on the base unit.
2) Check whether the operating environment of the Ethernet module is within the general specifications range of the CPU module.
3) Check whether the power supply capacity is sufficient or not.
4) Check whether the hardware of the Ethernet module, power supply module and CPU module are normal according to the manuals of the corresponding modules.
   If any module is faulty, please request your local Mitsubishi service center or representative to repair.
5) If the problem cannot be solved through the above steps, please consult your local Mitsubishi service center or representative, explaining the operation/communication conditions at error occurrence and the information stored in the error log area within the buffer memory of the Ethernet module.

11 - 39

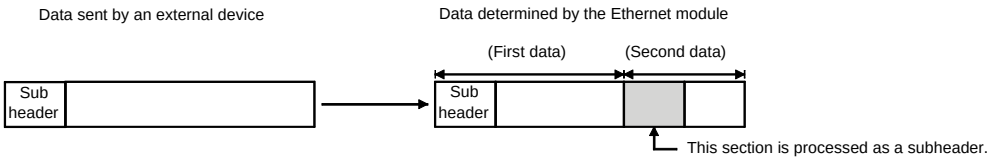
REMARK

Depending on the restrictions of the buffers of the local station and external device, data may be divided for communication.
Data received separately is restored (reassembled) by the Ethernet module to be communicated using the fixed buffer, random access buffer, etc. The received data is restored (reassembled) based on the data length in the communication data.
The Ethernet module performs the following processing if data among the communication data is incorrect.

(1) When communication is performed using fixed buffer (with procedure) and random buffer

(a) If the data length specified immediately after the subheader is smaller than the amount of text data received

1) The data immediately following the part of the text data equivalent to the data length specified immediately after the subheader will be regarded as the next message.
2) Since the header of each message will be a subheader, the Ethernet module performs processing according to the code in the subheader.
3) If the subheader contains a code that is not recognized by the Ethernet module, the Ethernet module will send a response notifying about abnormal completion to the external device.



At this point, the Ethernet module returns a response containing a code obtained by changing the most significant bit of the code processed as subheader to 1.
For example, if the subheader field of a command is 65h, the subheader of the response will become E5h.

11 - 40

(b) If the data length specified immediately after the subheader is larger than the amount of text data received
1) The Ethernet module waits for the reception of the remaining missing data.
2) If the remaining data could be received within the time allotted by the response monitoring timer, the Ethernet module performs processing according to the code in the subheader.
3) If the remaining data could not be received within the time allotted by the response monitoring timer, the Ethernet module performs the following processing.
• Sends the ABORT (RST) instruction to the external device and closes the line.
• Notifies about occurrence of an open error to the PLC CPU side via the open error detection signal (X18 = ON).
• Stores the error code in the open error code storage area. (The error code is not stored in the error log storage area.)

POINT

Designate the actual data size of the text field for "data length" specified in the application data field of a message sent from the external device to the Ethernet module.
The Ethernet module never sends a text whose size is different from the specified data length to the external device.

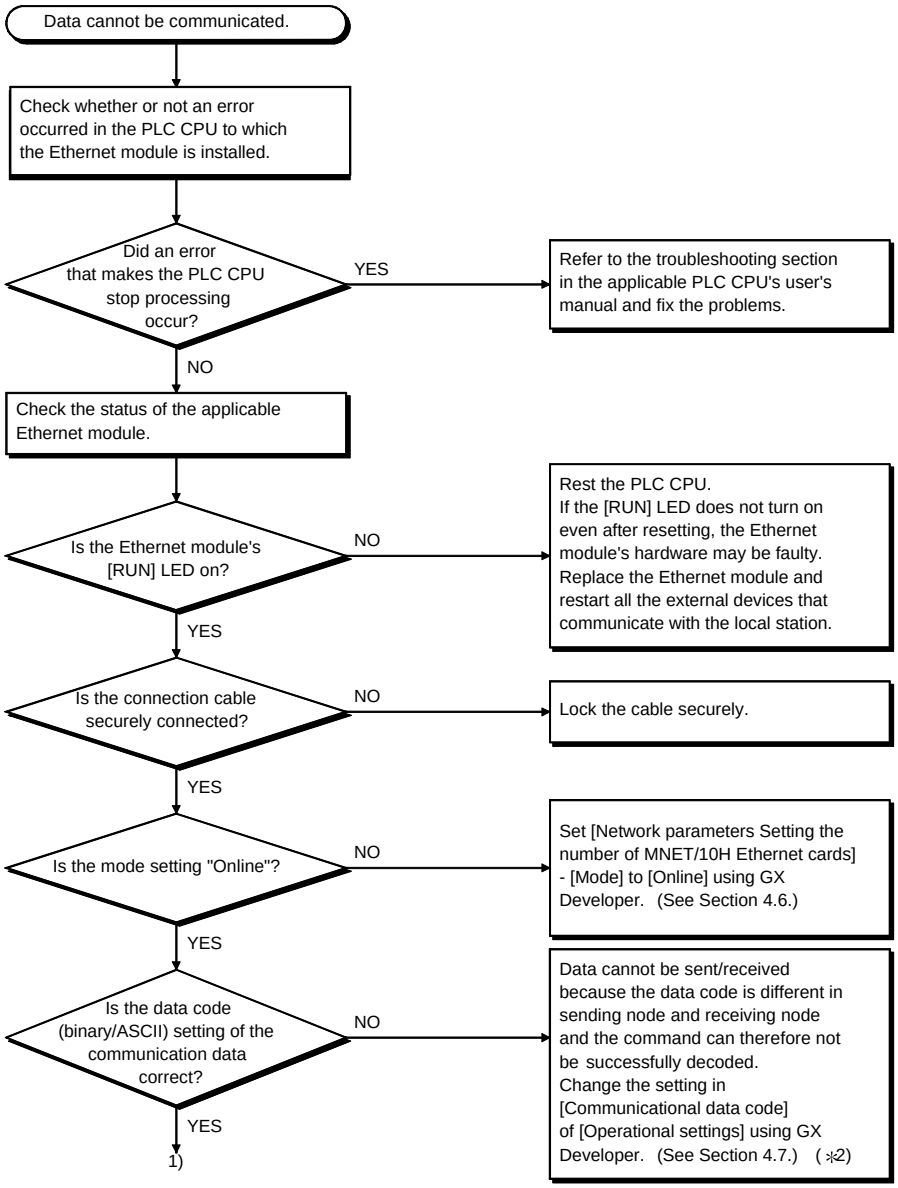
(2) When communication is performed using fixed buffer (non-procedure)
Since no message data length is specified in non-procedure communication, the data received is stored in the receive buffer area as is.
It is recommended to set up some method of checking that the data is received correctly. This can for instance be achieved by including the data length and data type code in the application data of the message, so that the number of bytes and data type of application data can be identified on the receiving side.

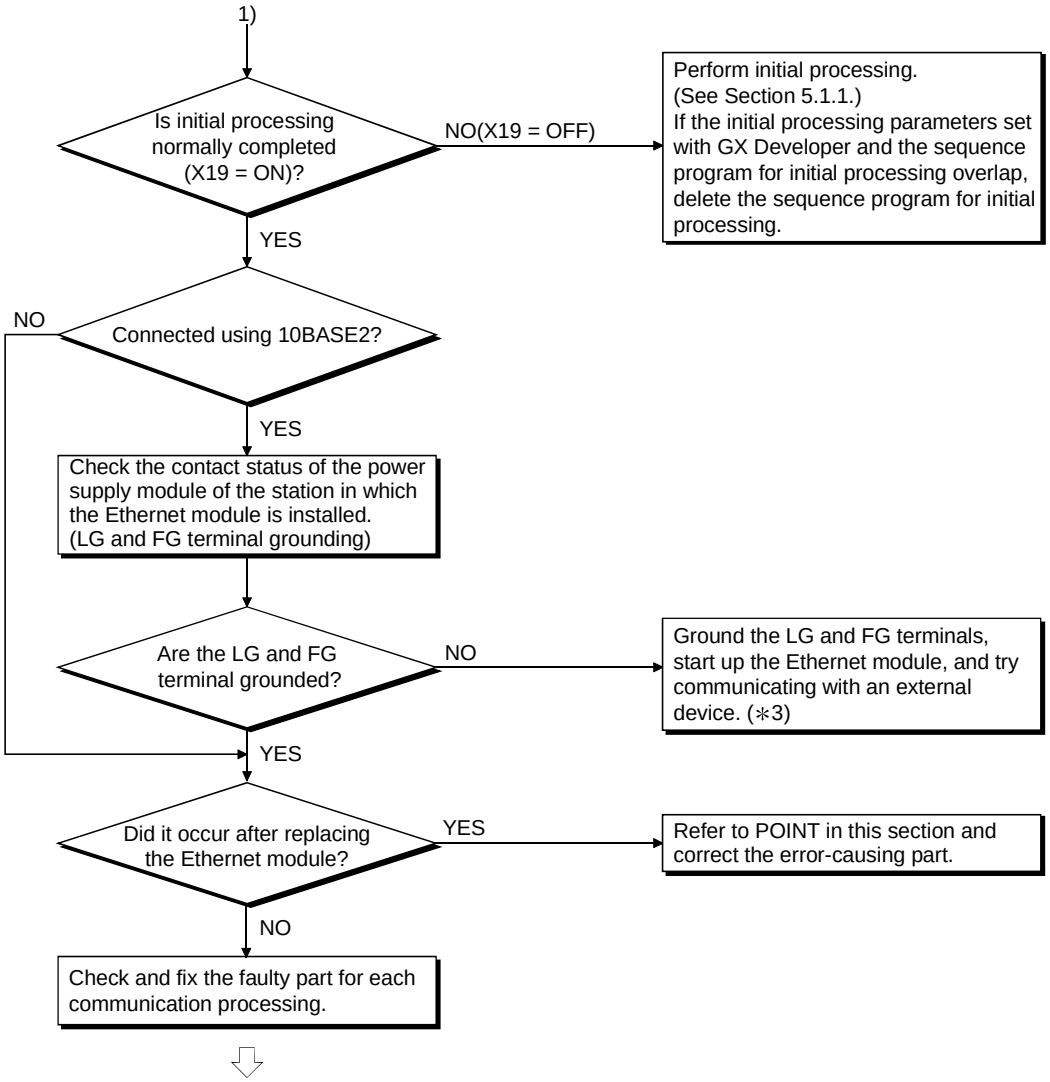
11.4 Troubleshooting Flowchart

This section explains some simple troubleshooting procedures when the Ethernet module and an external device have communication problems in a flowchart format. (*1)

POINT

If trouble occurs when using the Ethernet module, check the module status, error status and others for the Ethernet module using the Ethernet diagnostic function of GX Developer described in Section 11.2.


Data cannot be communicated.
Check whether or not an error occurred in the PLC CPU to which the Ethernet module is installed.
Did an error that makes the PLC CPU stop processing occur?
YES
Refer to the troubleshooting section in the applicable PLC CPU's user's manual and fix the problems.
NO
Check the status of the applicable Ethernet module.
Is the Ethernet module's [RUN] LED on?
NO
Rest the PLC CPU.
If the [RUN] LED does not turn on even after resetting, the Ethernet module's hardware may be faulty.
Replace the Ethernet module and restart all the external devices that communicate with the local station.
YES
Is the connection cable securely connected?
NO
Lock the cable securely.
YES
Is the mode setting "Online"?
NO
Set [Network parameters Setting the number of MNET/10H Ethernet cards] - [Mode] to [Online] using GX Developer. (See Section 4.6.)
YES
Is the data code (binary/ASCII) setting of the communication data correct?
NO
Data cannot be sent/received because the data code is different in sending node and receiving node and the command can therefore not be successfully decoded.
Change the setting in [Communicational data code] of [Operational settings] using GX Developer. (See Section 4.7.) (*2)
YES
1)

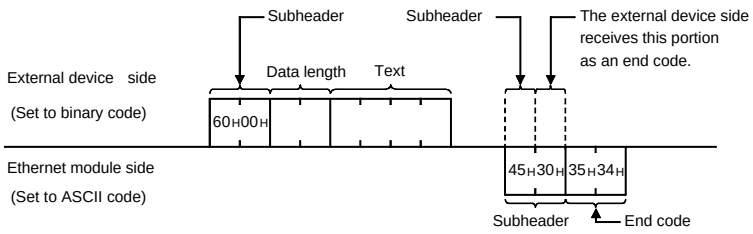

1)
Is initial processing normally completed (X19 = ON)?
NO(X19 = OFF)
Perform initial processing. (See Section 5.1.1.) If the initial processing parameters set with GX Developer and the sequence program for initial processing overlap, delete the sequence program for initial processing.
YES
NO
Connected using 10BASE2?
YES
Check the contact status of the power supply module of the station in which the Ethernet module is installed. (LG and FG terminal grounding)
Are the LG and FG terminal grounded?
NO
Ground the LG and FG terminals, start up the Ethernet module, and try communicating with an external device. (*3)
YES
Did it occur after replacing the Ethernet module?
YES
Refer to POINT in this section and correct the error-causing part.
NO
Check and fix the faulty part for each communication processing.

Sending error in fixed buffer communication: See Section 11.4.1.
Receiving error in fixed buffer communication: See Section 11.4.2.
Error in random access buffer communication: See Section 11.4.3.
Error in the MC protocol communication: See Section 11.4.4.
Send error during communication by e-mail: See Section 11.4.5.
Receive error during communication by e-mail: See Section 11.4.6.
Error in redundant system: See Section 11.4.7.

*1 See Section 11.1 when the I/O signal X1C (COM.ERR LED ON confirmation signal) is turned on or when the display LED COM.ERR (communication error detection display) lights up.
(Check the dedicated instruction control table for a processing during which an error occurred or the error code stored in the buffer memory, then check the contents of the error and take corrective actions by referring to Section 11.3.)

*2 Error codes not found in the error code list may be returned to the external device side if the communication data settings on the Ethernet module side (see Section 4.7) and the data code settings on the external device side are different.
The Ethernet module cannot decode commands correctly if data with different data codes is received. The Ethernet module returns error responses according to the communication data code settings.

[Example] When communication is performed using a fixed buffer



*3 If the LG and FG terminals of the power supply module of the station in which the Ethernet module is installed are not grounded, the communication line is closed (disconnected) due to the effects of noise, and as a result communication with an external device may not be performed.
Turn off the power to the station in which the Ethernet module is installed and ground the LG and FG terminals of the power supply module after referring to the section in the PLC CPU User's Manual that explains the wiring procedure.

POINT

(1) When the Ethernet module is replaced due to an error, reboot the following external devices and restart data communication:
- All external devices that communicated with the station whose Ethernet module was replaced.
- All external devices that communicated with other station's PLC CPU via a station whose Ethernet module was replaced.

(2) When connecting a device to the Ethernet module, see the following sections for the required devices and connection method:
Section 2.2: Devices Required for Network Configuration
Section 4.4: Connection Method Check

(3) If the Ethernet module could not receive messages sent from external devices frequently, check the values stored in the following buffer memory.

(a) Simultaneous transmission error detection count storage area (address: 18E$_H$ to 18F$_H$) and Error/End code storage area for each error log block (address: E5$_H$…)
When the error detection count number is high or when the error code C0C7$_H$ has been stored, high load may be applied to the Ethernet connection line to transfer data among connected devices.
To reduce the load to the Ethernet line, it is necessary to take corrective measures such as dividing the network or reducing the data transmission frequency.
Consult your network administrator and take appropriate measures.

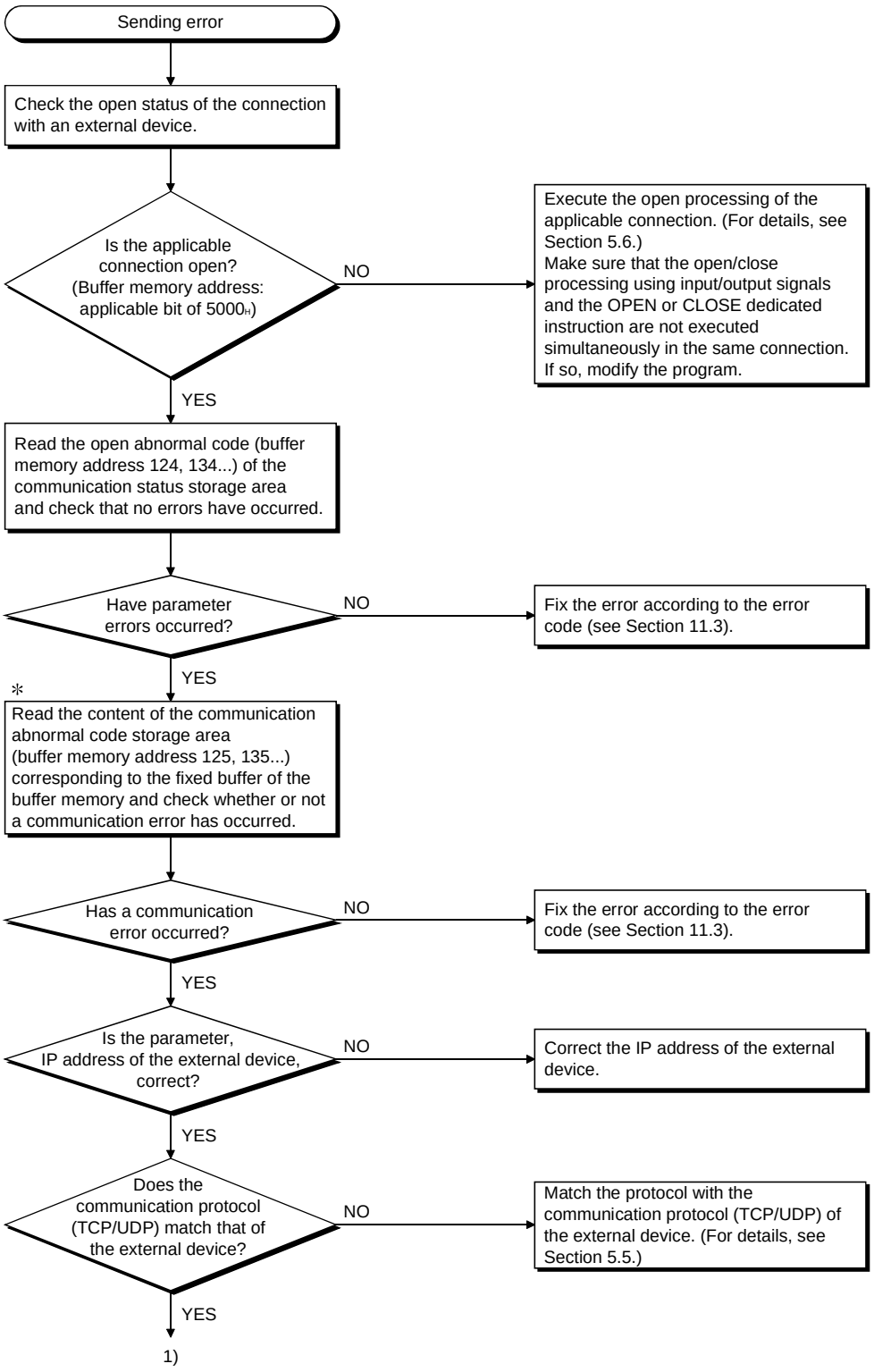
(b) TCP packet reception count storage area (address: 1B8$_H$, 1B9$_H$)
When data cannot be received even though TCP packet reception count has been renewed, set 8000$_H$ (Disable TCP Maximum Segment Size Option transmission) to the TCP Maximum Segment Transmission setting area (address: 1E$_H$) and execute re-initialization.
(For re-initialization, refer to Section 5.2.3.)

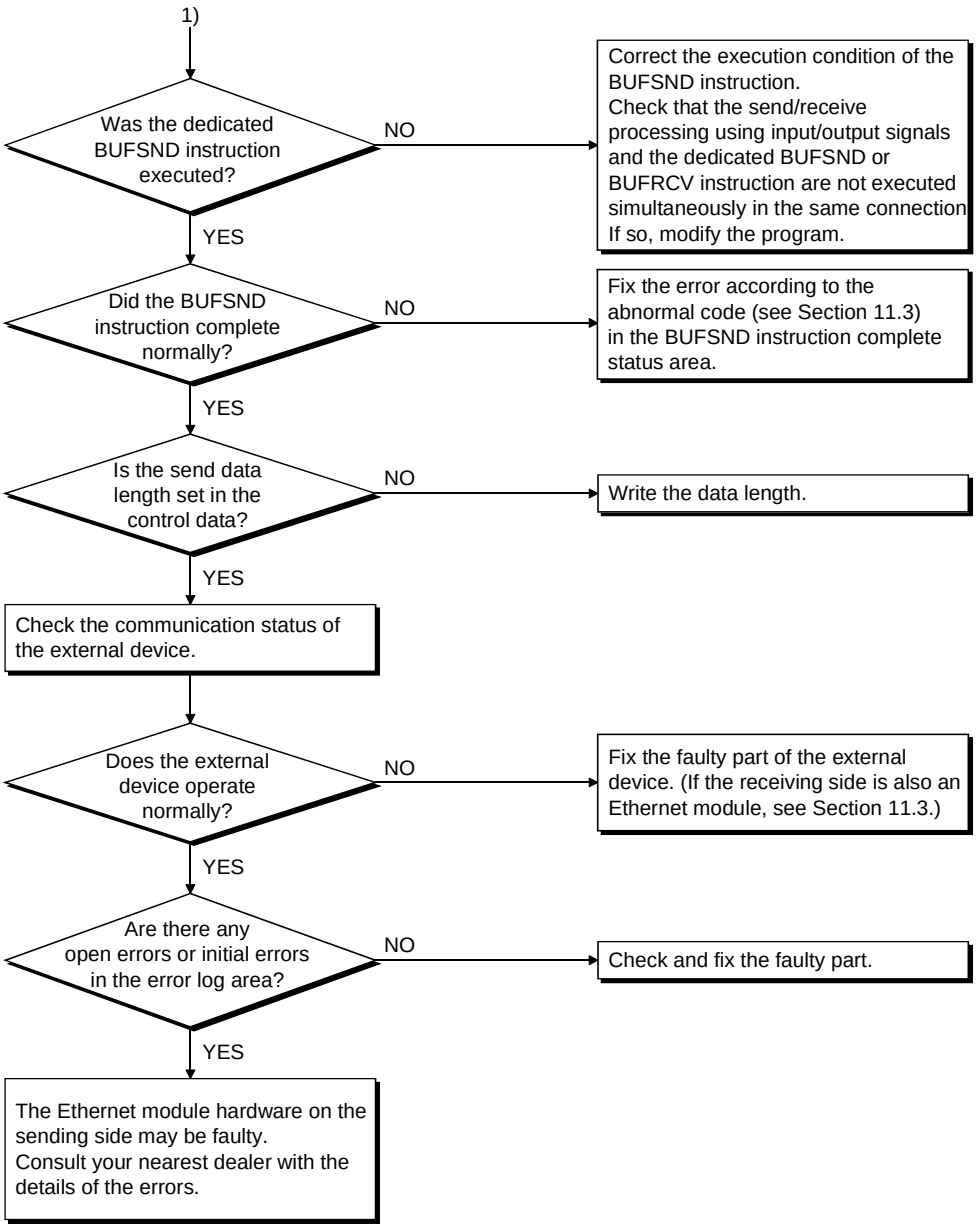
(4) All dedicated instructions must be executed online.
If any of the dedicated instructions is executed offline, no error will occur, but the execution of the dedicated instruction will not be completed.

11.4.1 Sending errors during fixed buffer communication (common to procedure exist and no procedure)

POINT

Check the abnormal code marked with "*" by referring to the complete status stored as the control data of the dedicated instruction.


Sending error
Check the open status of the connection with an external device.
Is the applicable connection open? (Buffer memory address: applicable bit of 5000H)
NO
Execute the open processing of the applicable connection. (For details, see Section 5.6.)
Make sure that the open/close processing using input/output signals and the OPEN or CLOSE dedicated instruction are not executed simultaneously in the same connection. If so, modify the program.
YES
Read the open abnormal code (buffer memory address 124, 134...) of the communication status storage area and check that no errors have occurred.
Have parameter errors occurred?
NO
Fix the error according to the error code (see Section 11.3).
YES
*
Read the content of the communication abnormal code storage area (buffer memory address 125, 135...) corresponding to the fixed buffer of the buffer memory and check whether or not a communication error has occurred.
Has a communication error occurred?
NO
Fix the error according to the error code (see Section 11.3).
YES
Is the parameter, IP address of the external device, correct?
NO
Correct the IP address of the external device.
YES
Does the communication protocol (TCP/UDP) match that of the external device?
NO
Match the protocol with the communication protocol (TCP/UDP) of the external device. (For details, see Section 5.5.)
YES
1)

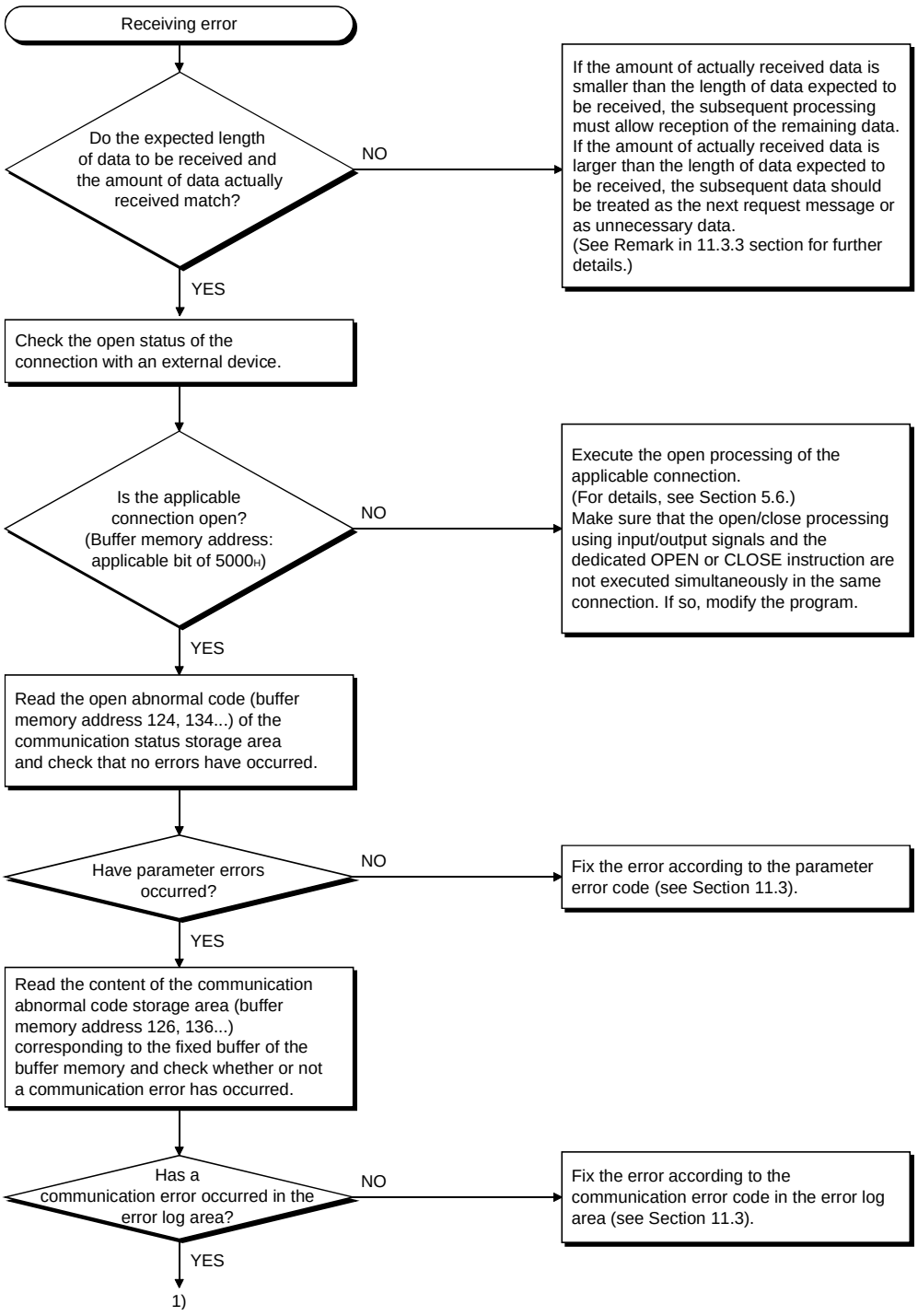

1)
Was the dedicated BUFSND instruction executed?
NO
Correct the execution condition of the BUFSND instruction.
Check that the send/receive processing using input/output signals and the dedicated BUFSND or BUFRCV instruction are not executed simultaneously in the same connection.
If so, modify the program.
YES
Did the BUFSND instruction complete normally?
NO
Fix the error according to the abnormal code (see Section 11.3) in the BUFSND instruction complete status area.
YES
Is the send data length set in the control data?
NO
Write the data length.
YES
Check the communication status of the external device.
Does the external device operate normally?
NO
Fix the faulty part of the external device. (If the receiving side is also an Ethernet module, see Section 11.3.)
YES
Are there any open errors or initial errors in the error log area?
NO
Check and fix the faulty part.
YES
The Ethernet module hardware on the sending side may be faulty.
Consult your nearest dealer with the details of the errors.

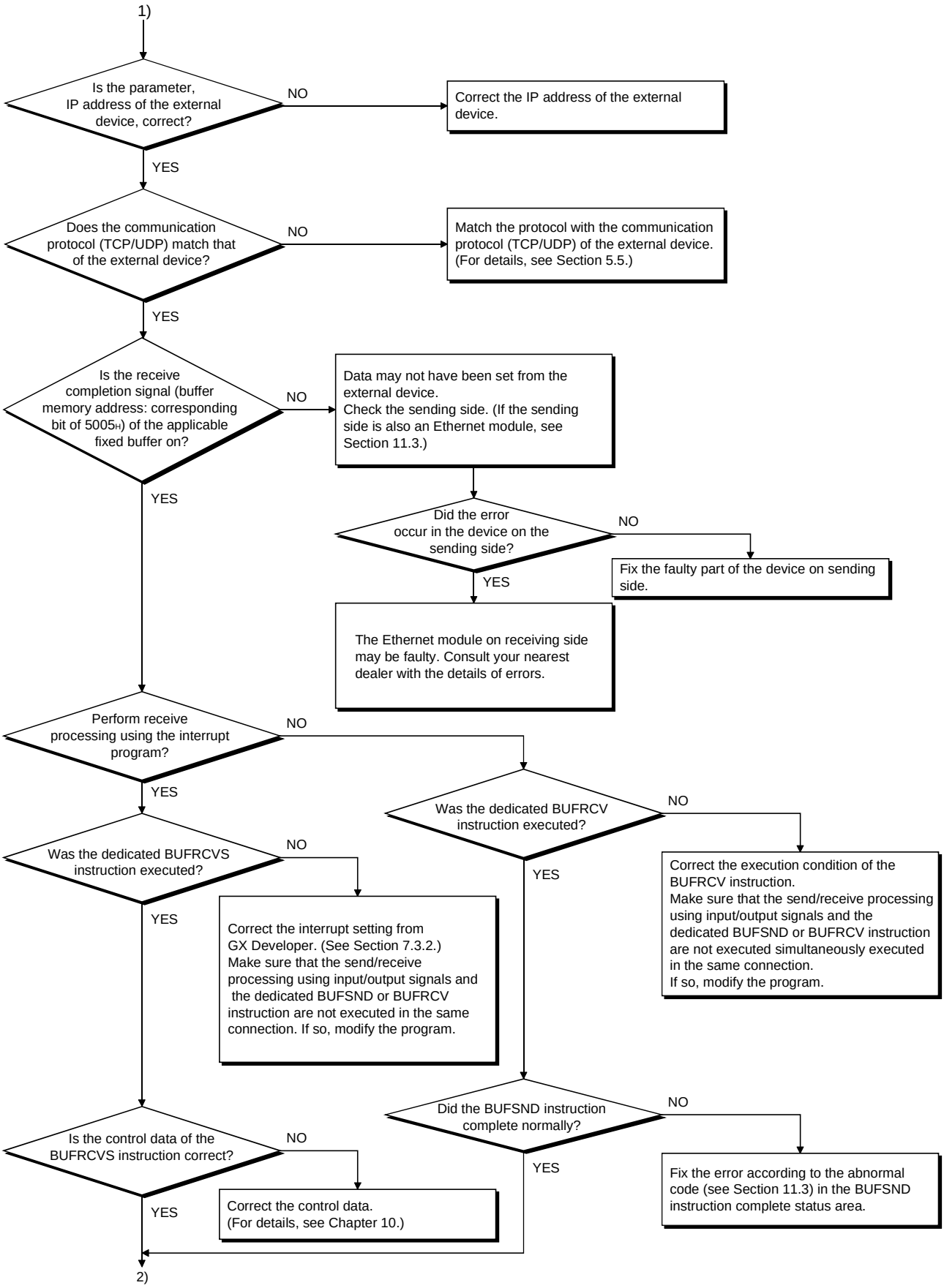
11 - 46

11.4.2 Receiving errors during fixed buffer communication (common to procedure exist and no procedure)

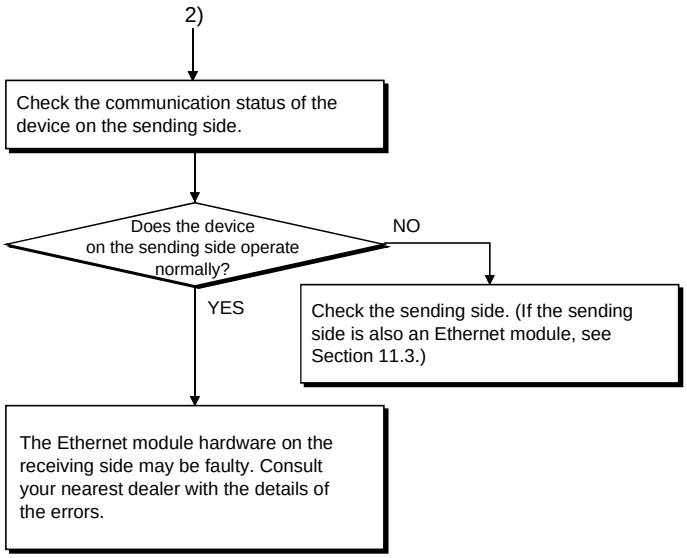
POINT

Perform error code reading while the connection is open.


Receiving error
Do the expected length of data to be received and the amount of data actually received match?
NO
If the amount of actually received data is smaller than the length of data expected to be received, the subsequent processing must allow reception of the remaining data.
If the amount of actually received data is larger than the length of data expected to be received, the subsequent data should be treated as the next request message or as unnecessary data.
(See Remark in 11.3.3 section for further details.)
YES
Check the open status of the connection with an external device.
Is the applicable connection open? (Buffer memory address: applicable bit of 5000H)
NO
Execute the open processing of the applicable connection.
(For details, see Section 5.6.)
Make sure that the open/close processing using input/output signals and the dedicated OPEN or CLOSE instruction are not executed simultaneously in the same connection. If so, modify the program.
YES
Read the open abnormal code (buffer memory address 124, 134...) of the communication status storage area and check that no errors have occurred.
Have parameter errors occurred?
NO
Fix the error according to the parameter error code (see Section 11.3).
YES
Read the content of the communication abnormal code storage area (buffer memory address 126, 136...) corresponding to the fixed buffer of the buffer memory and check whether or not a communication error has occurred.
Has a communication error occurred in the error log area?
NO
Fix the error according to the communication error code in the error log area (see Section 11.3).
YES
1)

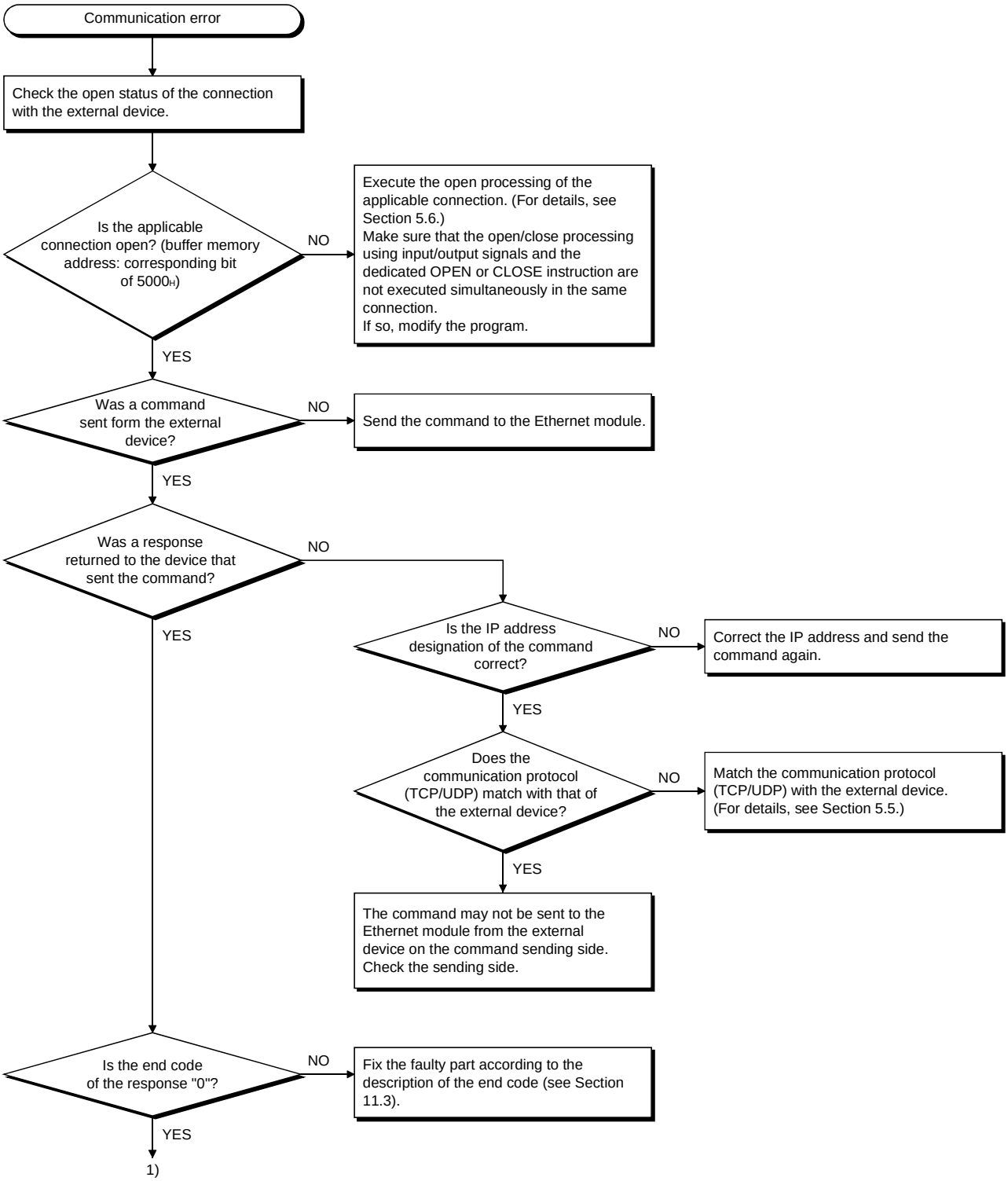

1)
Is the parameter, IP address of the external device, correct?
NO
Correct the IP address of the external device.
YES
Does the communication protocol (TCP/UDP) match that of the external device?
NO
Match the protocol with the communication protocol (TCP/UDP) of the external device. (For details, see Section 5.5.)
YES
Is the receive completion signal (buffer memory address: corresponding bit of 5005H) of the applicable fixed buffer on?
NO
Data may not have been set from the external device. Check the sending side. (If the sending side is also an Ethernet module, see Section 11.3.)
Did the error occur in the device on the sending side?
NO
Fix the faulty part of the device on sending side.
YES
The Ethernet module on receiving side may be faulty. Consult your nearest dealer with the details of errors.
YES
Perform receive processing using the interrupt program?
NO
YES
Was the dedicated BUFRCVS instruction executed?
NO
Correct the interrupt setting from GX Developer. (See Section 7.3.2.) Make sure that the send/receive processing using input/output signals and the dedicated BUFSND or BUFRCV instruction are not executed in the same connection. If so, modify the program.
YES
Was the dedicated BUFRCV instruction executed?
NO
Correct the execution condition of the BUFRCV instruction. Make sure that the send/receive processing using input/output signals and the dedicated BUFSND or BUFRCV instruction are not executed simultaneously executed in the same connection. If so, modify the program.
YES
Is the control data of the BUFRCVS instruction correct?
NO
Correct the control data. (For details, see Chapter 10.)
YES
Did the BUFSND instruction complete normally?
NO
Fix the error according to the abnormal code (see Section 11.3) in the BUFSND instruction complete status area.
YES
2)

11 - 48

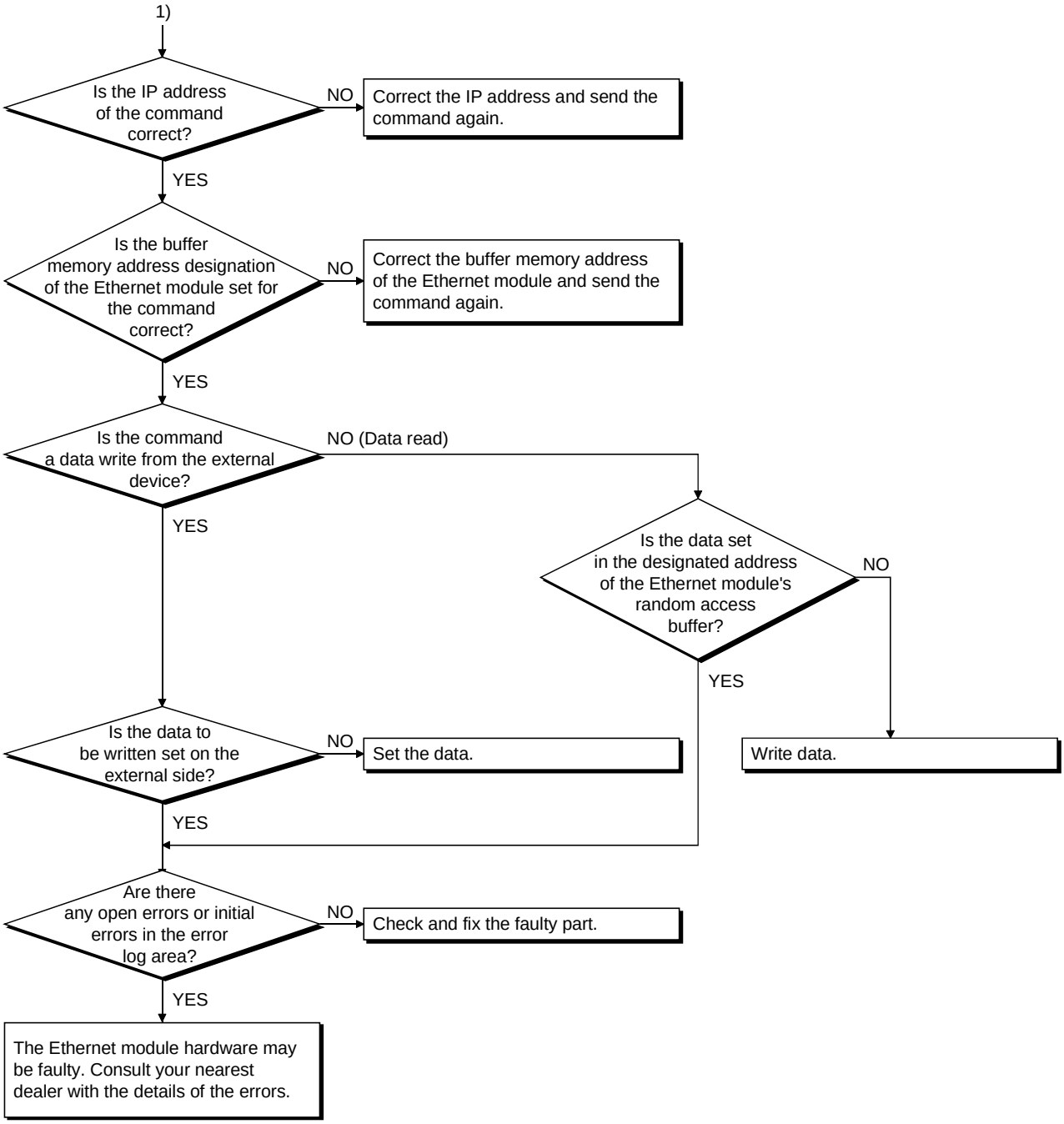

2)
Check the communication status of the device on the sending side.
Does the device on the sending side operate normally?
NO
YES
Check the sending side. (If the sending side is also an Ethernet module, see Section 11.3.)
The Ethernet module hardware on the receiving side may be faulty. Consult your nearest dealer with the details of the errors.

11 - 49

11.4.3 Errors during random access buffer communication

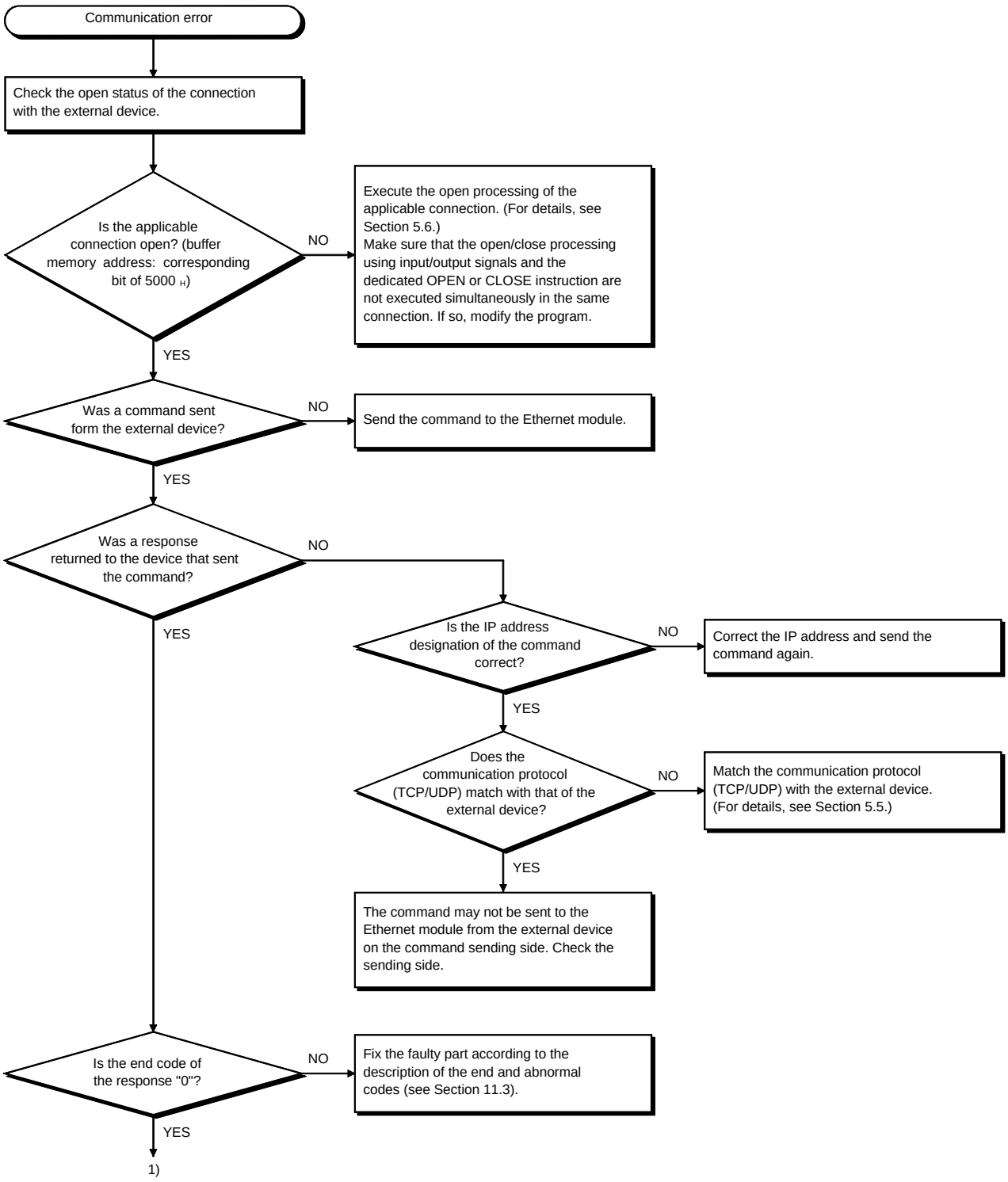

Communication error
Check the open status of the connection with the external device.
Is the applicable connection open? (buffer memory address: corresponding bit of 5000H)
NO
Execute the open processing of the applicable connection. (For details, see Section 5.6.)
Make sure that the open/close processing using input/output signals and the dedicated OPEN or CLOSE instruction are not executed simultaneously in the same connection.
If so, modify the program.
YES
Was a command sent form the external device?
NO
Send the command to the Ethernet module.
YES
Was a response returned to the device that sent the command?
NO
YES
Is the IP address designation of the command correct?
NO
Correct the IP address and send the command again.
YES
Does the communication protocol (TCP/UDP) match with that of the external device?
NO
Match the communication protocol (TCP/UDP) with the external device. (For details, see Section 5.5.)
YES
The command may not be sent to the Ethernet module from the external device on the command sending side. Check the sending side.
Is the end code of the response "0"?
NO
Fix the faulty part according to the description of the end code (see Section 11.3).
YES
1)

11 - 50

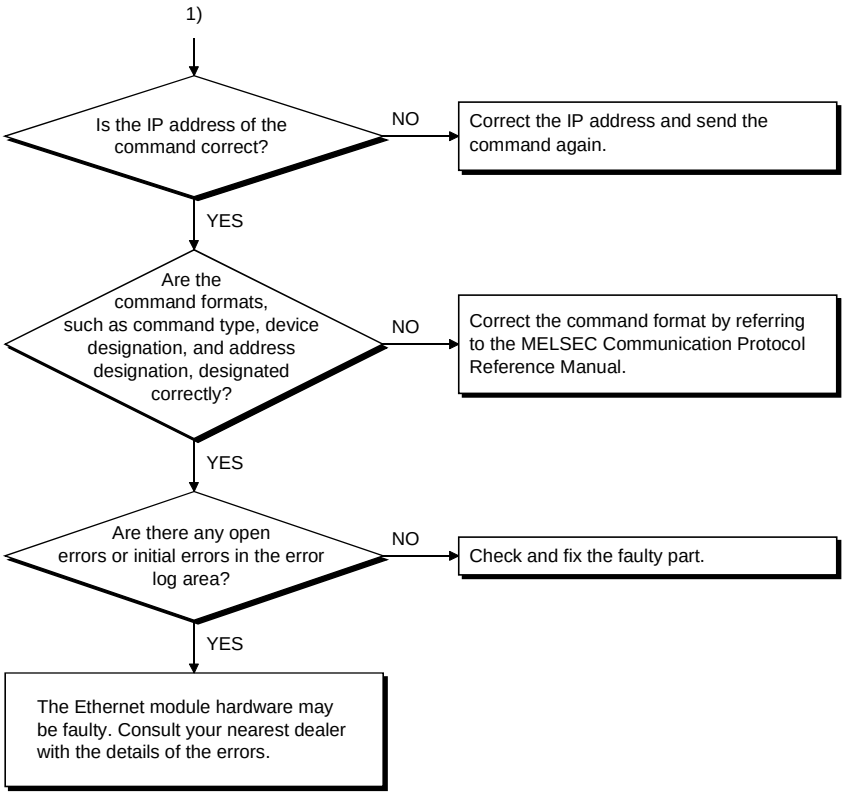

1)
Is the IP address of the command correct?
NO
Correct the IP address and send the command again.
YES
Is the buffer memory address designation of the Ethernet module set for the command correct?
NO
Correct the buffer memory address of the Ethernet module and send the command again.
YES
Is the command a data write from the external device?
NO (Data read)
YES
Is the data set in the designated address of the Ethernet module's random access buffer?
NO
YES
Write data.
Is the data to be written set on the external side?
NO
Set the data.
YES
Are there any open errors or initial errors in the error log area?
NO
Check and fix the faulty part.
YES
The Ethernet module hardware may be faulty. Consult your nearest dealer with the details of the errors.

11 - 51

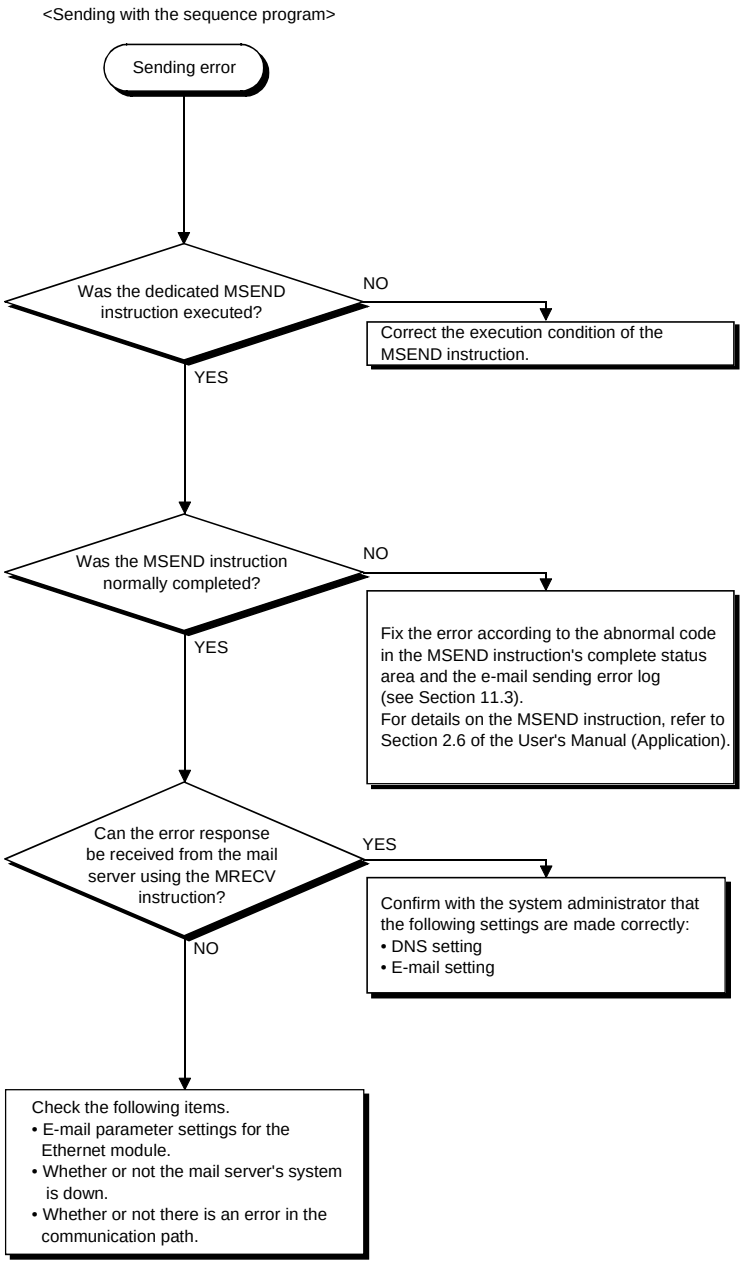
11.4.4 Errors in communication using the MC protocol


Communication error
Check the open status of the connection with the external device.
Is the applicable connection open? (buffer memory address: corresponding bit of 5000 H)
NO
Execute the open processing of the applicable connection. (For details, see Section 5.6.)
Make sure that the open/close processing using input/output signals and the dedicated OPEN or CLOSE instruction are not executed simultaneously in the same connection. If so, modify the program.
YES
Was a command sent form the external device?
NO
Send the command to the Ethernet module.
YES
Was a response returned to the device that sent the command?
NO
YES
Is the IP address designation of the command correct?
NO
Correct the IP address and send the command again.
YES
Does the communication protocol (TCP/UDP) match with that of the external device?
NO
Match the communication protocol (TCP/UDP) with the external device. (For details, see Section 5.5.)
YES
The command may not be sent to the Ethernet module from the external device on the command sending side. Check the sending side.
Is the end code of the response "0"?
NO
Fix the faulty part according to the description of the end and abnormal codes (see Section 11.3).
YES
1)

11 - 52

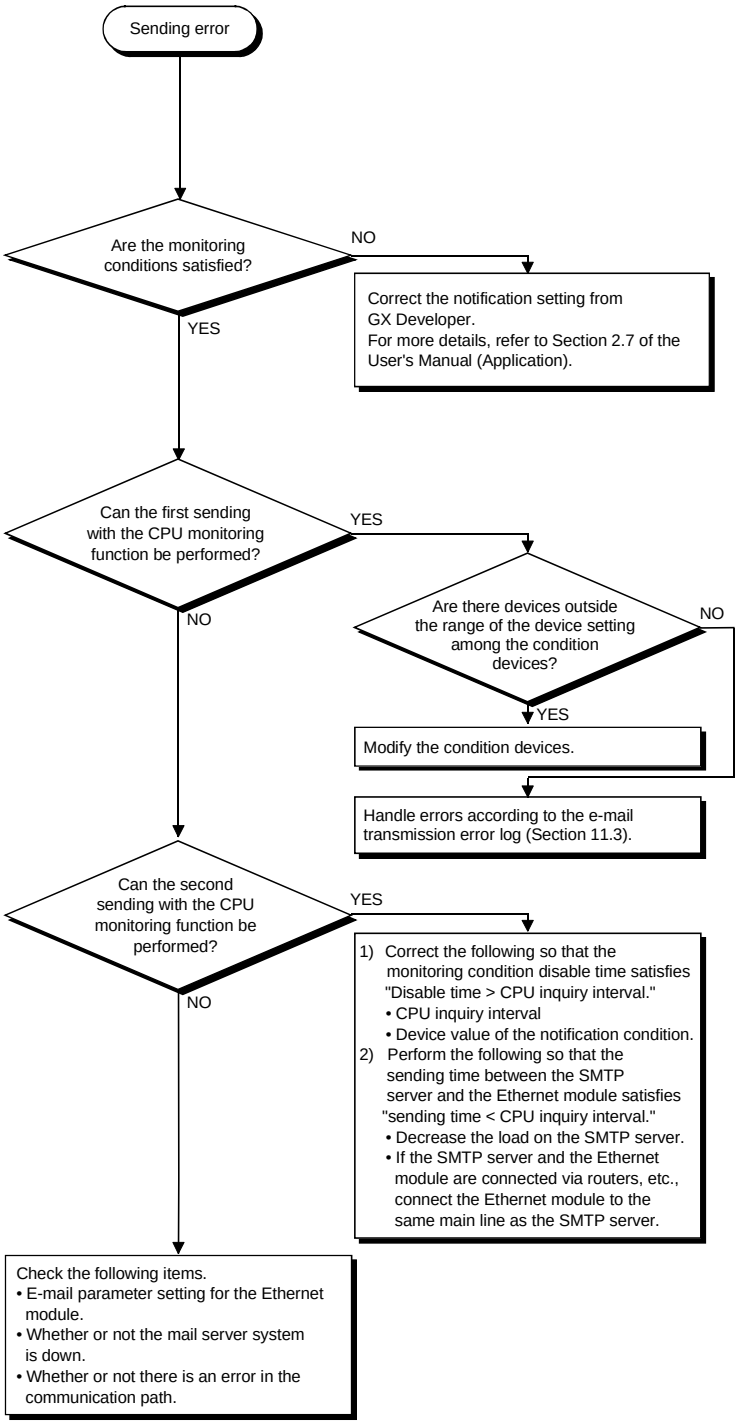

1)
Is the IP address of the command correct?
NO
Correct the IP address and send the command again.
YES
Are the command formats, such as command type, device designation, and address designation, designated correctly?
NO
Correct the command format by referring to the MELSEC Communication Protocol Reference Manual.
YES
Are there any open errors or initial errors in the error log area?
NO
Check and fix the faulty part.
YES
The Ethernet module hardware may be faulty. Consult your nearest dealer with the details of the errors.

11.4.5 Sending errors during e-mail communication


<Sending with the sequence program>
Sending error
Was the dedicated MSEND instruction executed?
NO
Correct the execution condition of the MSEND instruction.
YES
Was the MSEND instruction normally completed?
NO
Fix the error according to the abnormal code in the MSEND instruction's complete status area and the e-mail sending error log (see Section 11.3).
For details on the MSEND instruction, refer to Section 2.6 of the User's Manual (Application).
YES
Can the error response be received from the mail server using the MRECV instruction?
YES
Confirm with the system administrator that the following settings are made correctly:
• DNS setting
• E-mail setting
NO
Check the following items.
• E-mail parameter settings for the Ethernet module.
• Whether or not the mail server's system is down.
• Whether or not there is an error in the communication path.

11 - 54

<Sending with the CPU monitoring function>


Sending error
Are the monitoring conditions satisfied?
NO
Correct the notification setting from GX Developer.
For more details, refer to Section 2.7 of the User's Manual (Application).
YES
Can the first sending with the CPU monitoring function be performed?
YES
Are there devices outside the range of the device setting among the condition devices?
NO
YES
Modify the condition devices.
Handle errors according to the e-mail transmission error log (Section 11.3).
NO
Can the second sending with the CPU monitoring function be performed?
YES
1) Correct the following so that the monitoring condition disable time satisfies "Disable time > CPU inquiry interval."
• CPU inquiry interval
• Device value of the notification condition.
2) Perform the following so that the sending time between the SMTP server and the Ethernet module satisfies "sending time < CPU inquiry interval."
• Decrease the load on the SMTP server.
• If the SMTP server and the Ethernet module are connected via routers, etc., connect the Ethernet module to the same main line as the SMTP server.
NO
Check the following items.
• E-mail parameter setting for the Ethernet module.
• Whether or not the mail server system is down.
• Whether or not there is an error in the communication path.

POINT

Check the following items before using the e-mail function:

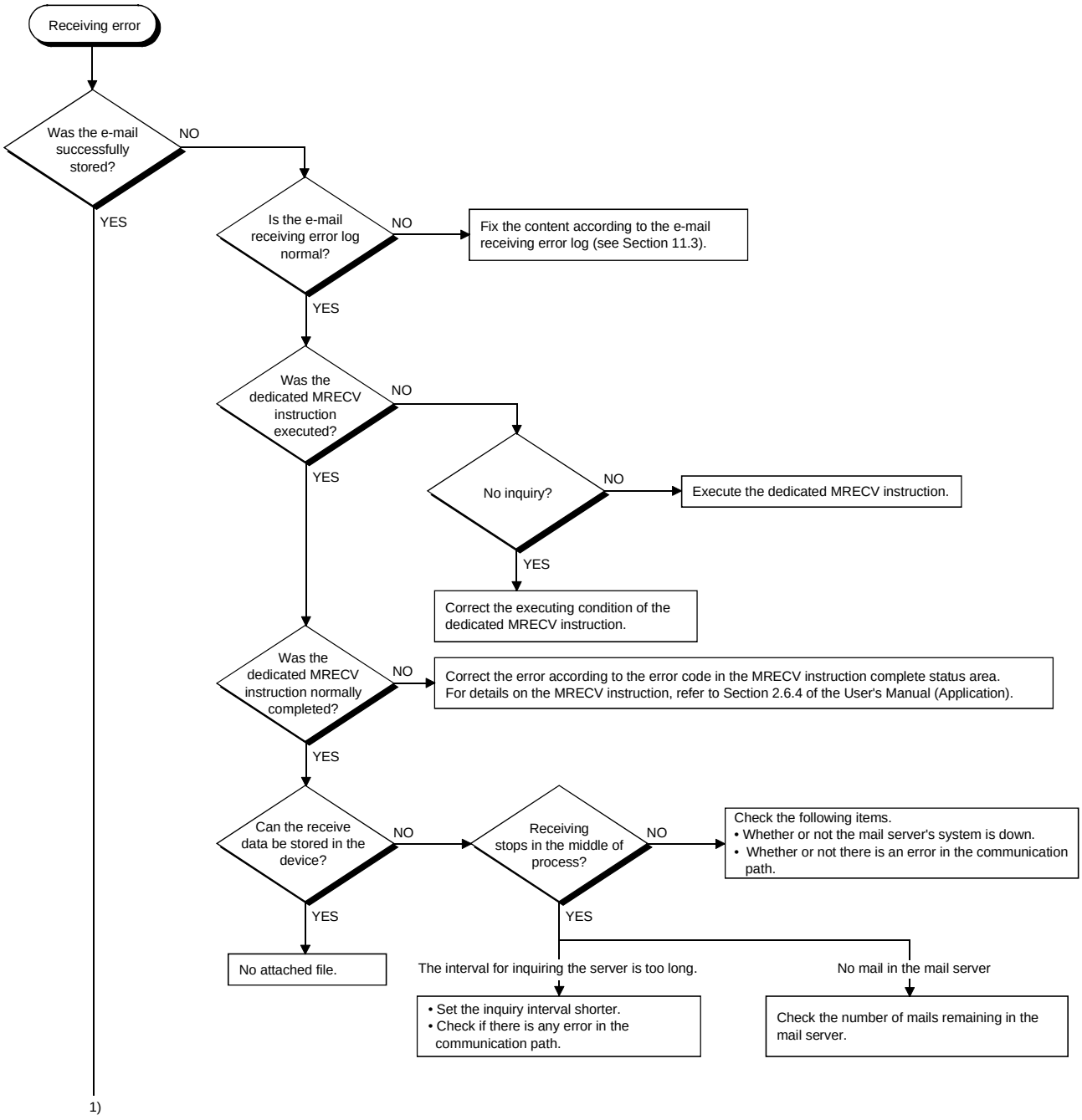
(1) System configuration and environment:
Section 2.2 of the User's Manual (Application)

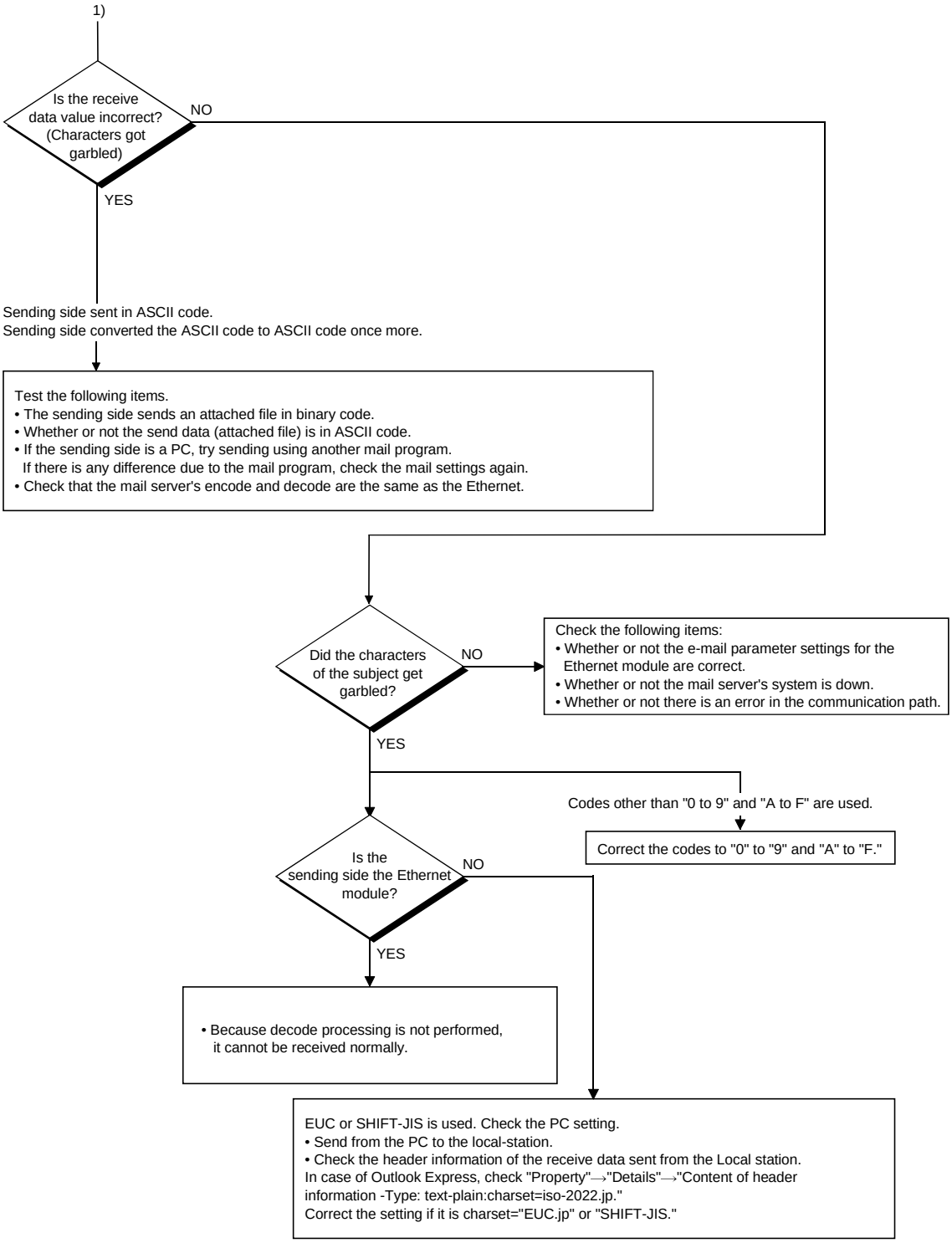
(2) Precautions for using the E-mail function:
Section 2.3 of the User's Manual (Application)

(3) The e-mail settings from GX Developer:
Section 2.5 of the User's Manual (Application)

11 - 55

11.4.6 Receiving errors during e-mail communication


Receiving error
Was the e-mail successfully stored?
NO
YES
Is the e-mail receiving error log normal?
NO
Fix the content according to the e-mail receiving error log (see Section 11.3).
YES
Was the dedicated MRECV instruction executed?
NO
YES
No inquiry?
NO
Execute the dedicated MRECV instruction.
YES
Correct the executing condition of the dedicated MRECV instruction.
Was the dedicated MRECV instruction normally completed?
NO
Correct the error according to the error code in the MRECV instruction complete status area.
For details on the MRECV instruction, refer to Section 2.6.4 of the User's Manual (Application).
YES
Can the receive data be stored in the device?
NO
Receiving stops in the middle of process?
NO
Check the following items.
• Whether or not the mail server's system is down.
• Whether or not there is an error in the communication path.
YES
YES
No attached file.
The interval for inquiring the server is too long.
• Set the inquiry interval shorter.
• Check if there is any error in the communication path.
No mail in the mail server
Check the number of mails remaining in the mail server.
1)

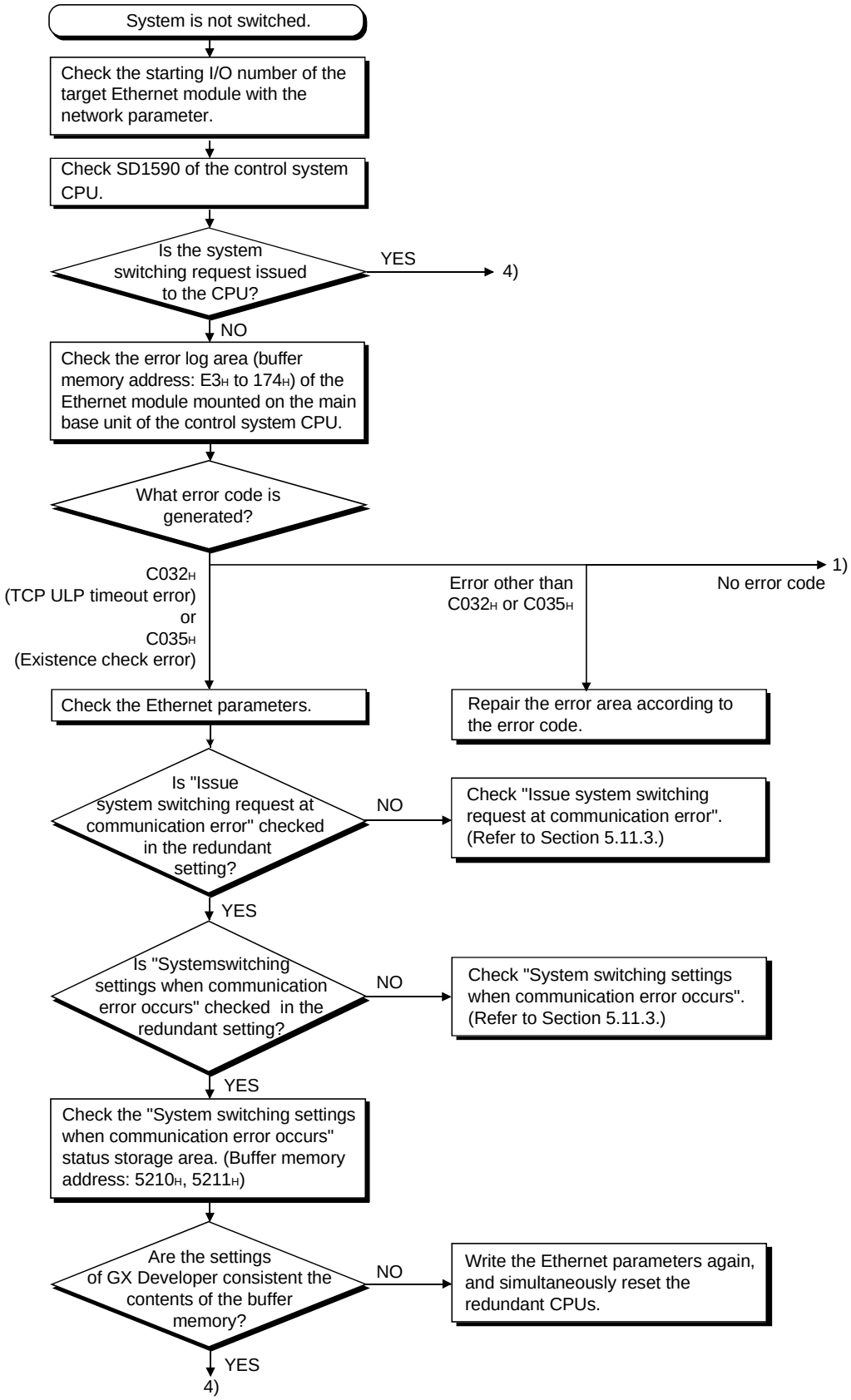

1)
Is the receive data value incorrect? (Characters got garbled)
NO
YES
Sending side sent in ASCII code.
Sending side converted the ASCII code to ASCII code once more.
Test the following items.
• The sending side sends an attached file in binary code.
• Whether or not the send data (attached file) is in ASCII code.
• If the sending side is a PC, try sending using another mail program.
If there is any difference due to the mail program, check the mail settings again.
• Check that the mail server's encode and decode are the same as the Ethernet.
Did the characters of the subject get garbled?
NO
Check the following items:
• Whether or not the e-mail parameter settings for the Ethernet module are correct.
• Whether or not the mail server's system is down.
• Whether or not there is an error in the communication path.
YES
Codes other than "0 to 9" and "A to F" are used.
Correct the codes to "0" to "9" and "A" to "F."
Is the sending side the Ethernet module?
NO
YES
• Because decode processing is not performed, it cannot be received normally.
EUC or SHIFT-JIS is used. Check the PC setting.
• Send from the PC to the local-station.
• Check the header information of the receive data sent from the Local station.
In case of Outlook Express, check "Property"→"Details"→"Content of header information -Type: text-plain:charset=iso-2022.jp."
Correct the setting if it is charset="EUC.jp" or "SHIFT-JIS."

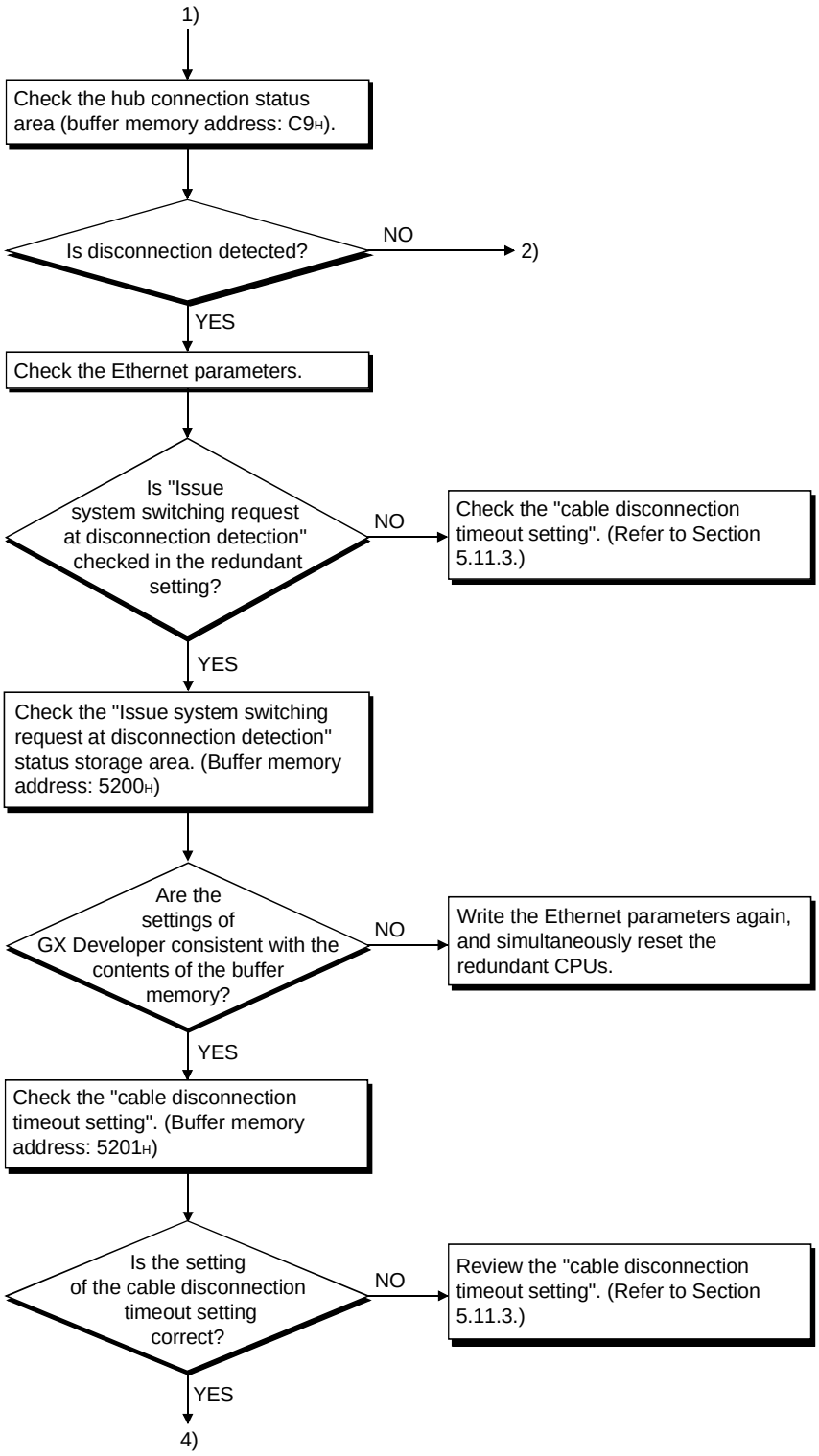
11 - 57

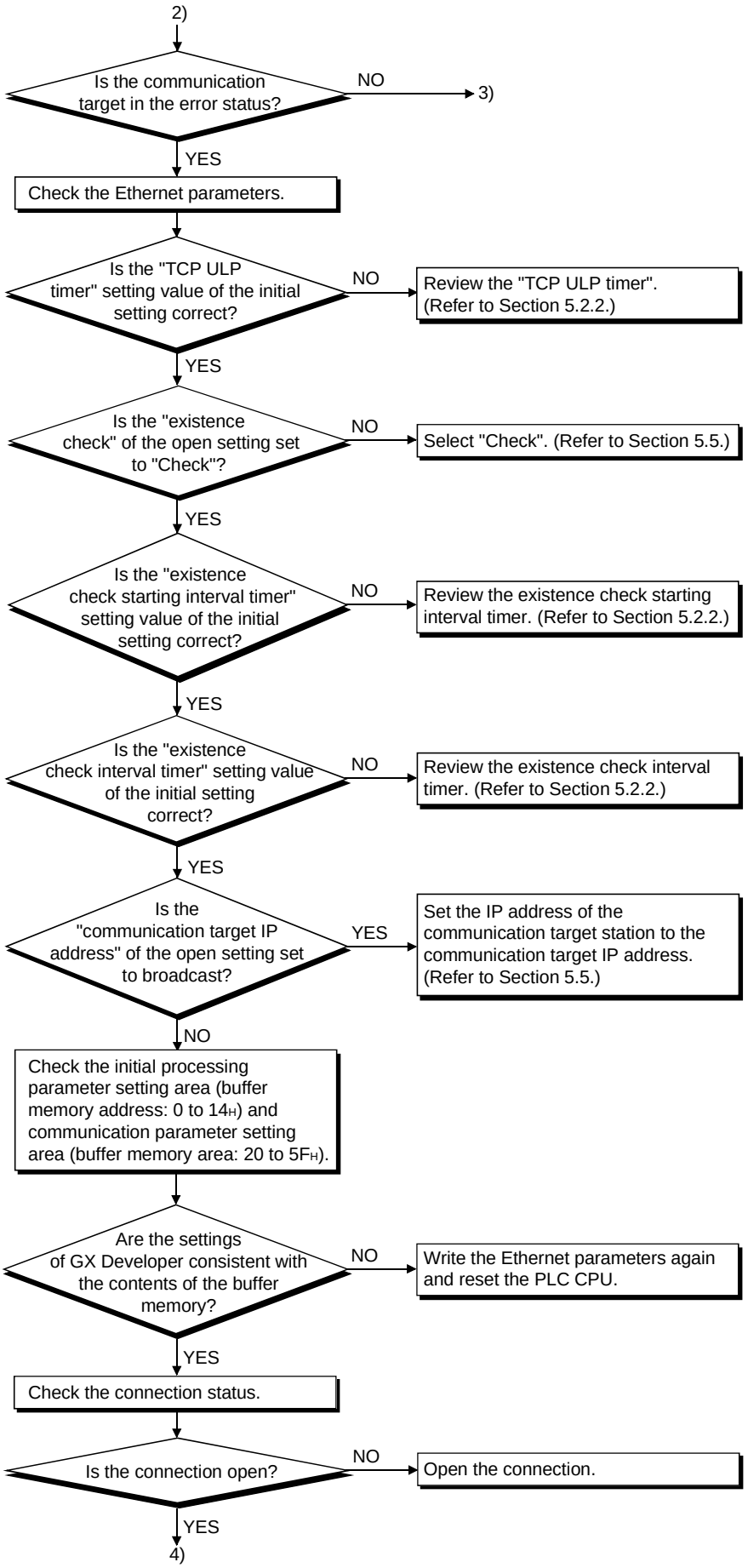
11.4.7 Error in redundant system

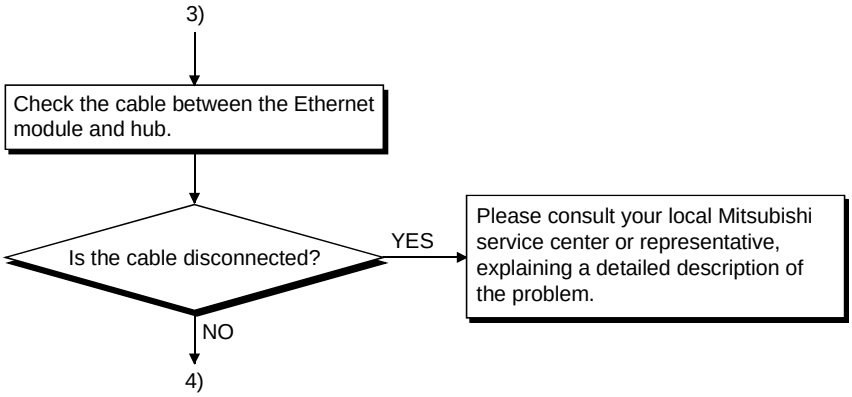
(1) System switching error

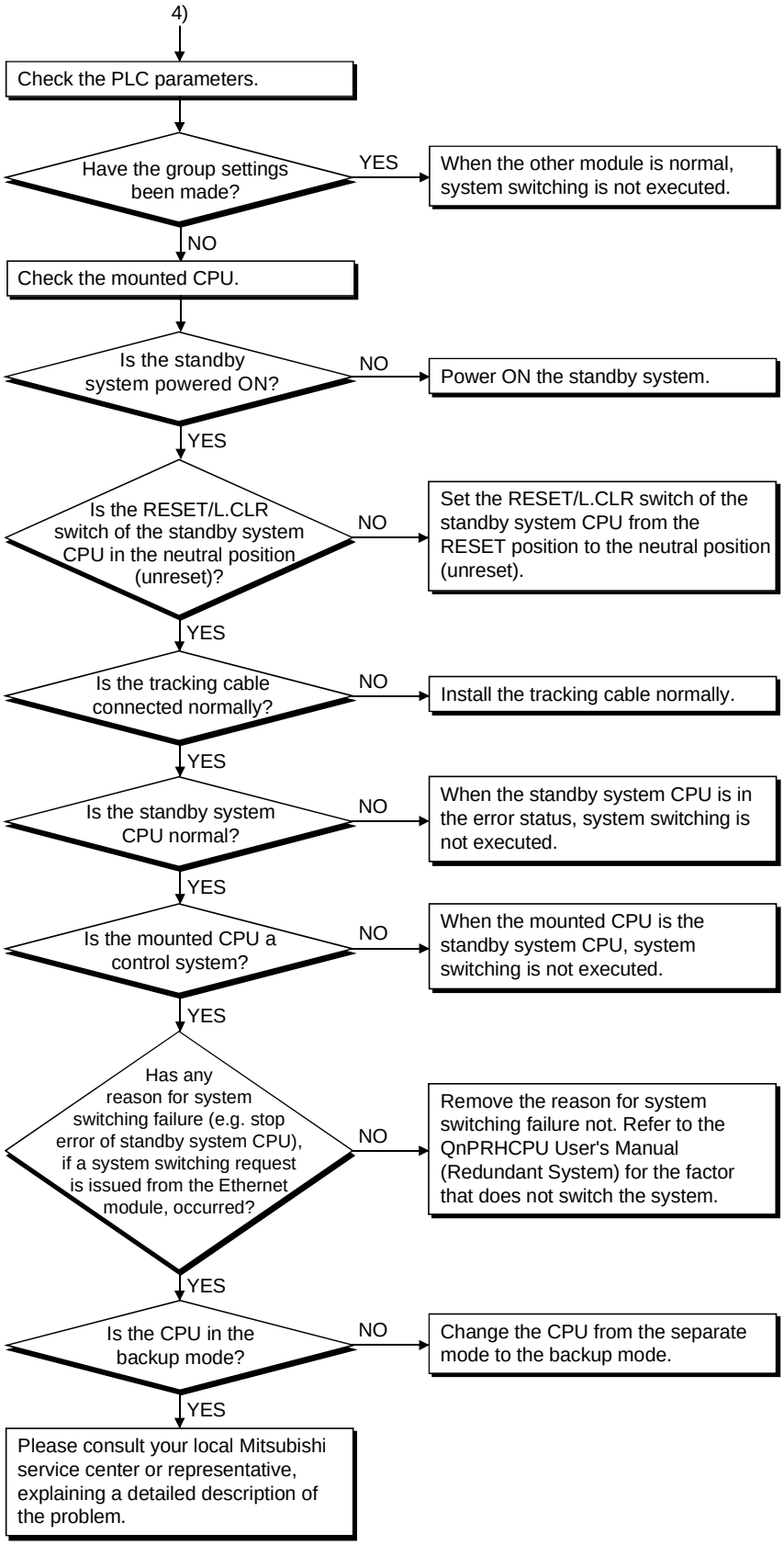
The following shows a troubleshooting flowchart for the case where system switching does not occur at communication error or disconnection detection.


System is not switched.
Check the starting I/O number of the target Ethernet module with the network parameter.
Check SD1590 of the control system CPU.
Is the system switching request issued to the CPU?
YES
4)
NO
Check the error log area (buffer memory address: E3H to 174H) of the Ethernet module mounted on the main base unit of the control system CPU.
What error code is generated?
C032H (TCP ULP timeout error) or C035H (Existence check error)
Error other than C032H or C035H
No error code
1)
Check the Ethernet parameters.
Repair the error area according to the error code.
Is "Issue system switching request at communication error" checked in the redundant setting?
NO
Check "Issue system switching request at communication error". (Refer to Section 5.11.3.)
YES
Is "Systemswitching settings when communication error occurs" checked in the redundant setting?
NO
Check "System switching settings when communication error occurs". (Refer to Section 5.11.3.)
YES
Check the "System switching settings when communication error occurs" status storage area. (Buffer memory address: 5210H, 5211H)
Are the settings of GX Developer consistent the contents of the buffer memory?
NO
Write the Ethernet parameters again, and simultaneously reset the redundant CPUs.
YES
4)


1)
Check the hub connection status area (buffer memory address: C9H).
Is disconnection detected?
NO
2)
YES
Check the Ethernet parameters.
Is "Issue system switching request at disconnection detection" checked in the redundant setting?
NO
Check the "cable disconnection timeout setting". (Refer to Section 5.11.3.)
YES
Check the "Issue system switching request at disconnection detection" status storage area. (Buffer memory address: 5200H)
Are the settings of GX Developer consistent with the contents of the buffer memory?
NO
Write the Ethernet parameters again, and simultaneously reset the redundant CPUs.
YES
Check the "cable disconnection timeout setting". (Buffer memory address: 5201H)
Is the setting of the cable disconnection timeout setting correct?
NO
Review the "cable disconnection timeout setting". (Refer to Section 5.11.3.)
YES
4)

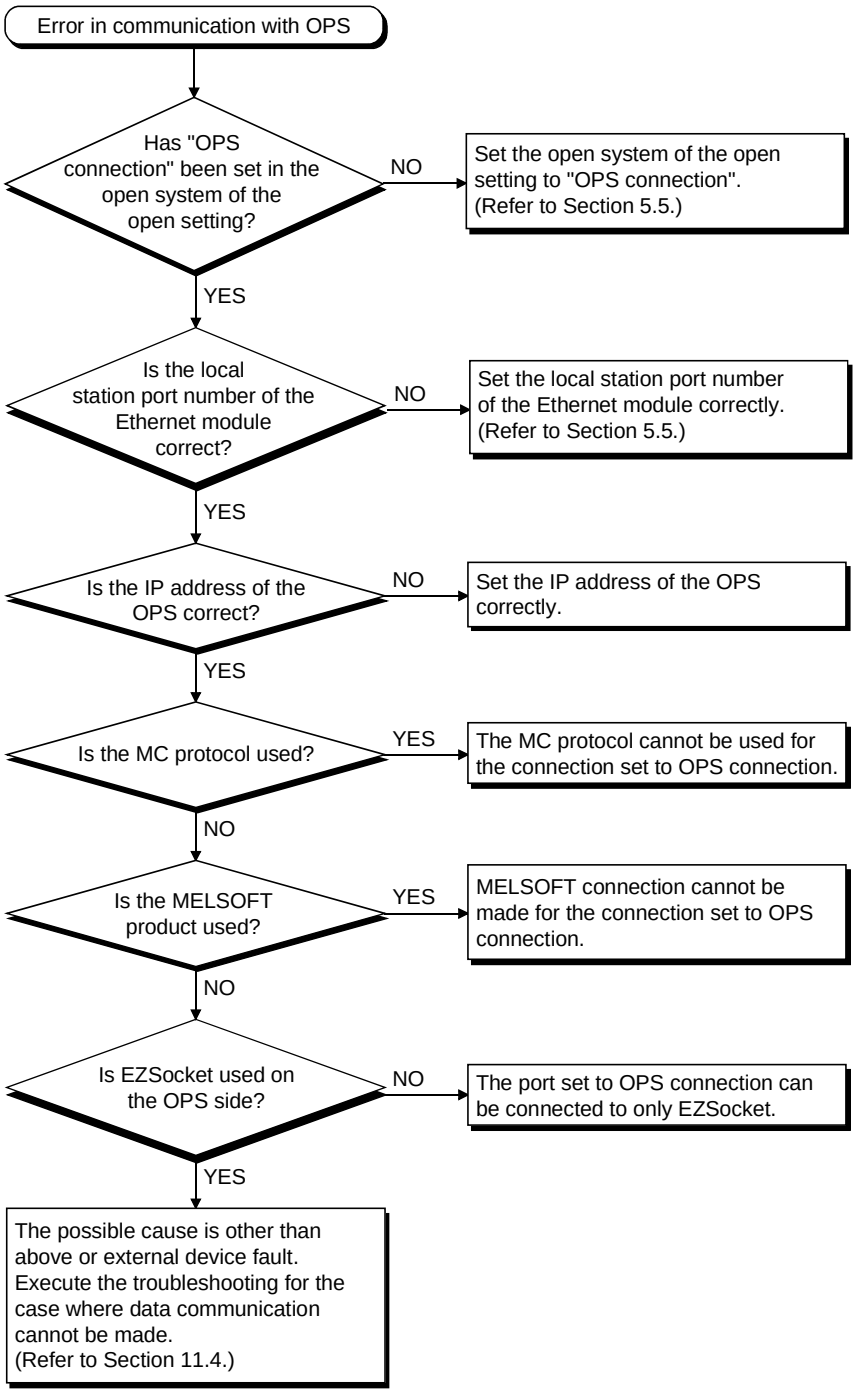

2)
Is the communication target in the error status?
NO
3)
YES
Check the Ethernet parameters.
Is the "TCP ULP timer" setting value of the initial setting correct?
NO
Review the "TCP ULP timer". (Refer to Section 5.2.2.)
YES
Is the "existence check" of the open setting set to "Check"?
NO
Select "Check". (Refer to Section 5.5.)
YES
Is the "existence check starting interval timer" setting value of the initial setting correct?
NO
Review the existence check starting interval timer. (Refer to Section 5.2.2.)
YES
Is the "existence check interval timer" setting value of the initial setting correct?
NO
Review the existence check interval timer. (Refer to Section 5.2.2.)
YES
Is the "communication target IP address" of the open setting set to broadcast?
YES
Set the IP address of the communication target station to the communication target IP address. (Refer to Section 5.5.)
NO
Check the initial processing parameter setting area (buffer memory address: 0 to 14H) and communication parameter setting area (buffer memory area: 20 to 5FH).
Are the settings of GX Developer consistent with the contents of the buffer memory?
NO
Write the Ethernet parameters again and reset the PLC CPU.
YES
Check the connection status.
Is the connection open?
NO
Open the connection.
YES
4)


3)
Check the cable between the Ethernet module and hub.
Is the cable disconnected?
YES
Please consult your local Mitsubishi service center or representative, explaining a detailed description of the problem.
NO
4)


4)
Check the PLC parameters.
Have the group settings been made?
YES
When the other module is normal, system switching is not executed.
NO
Check the mounted CPU.
Is the standby system powered ON?
NO
Power ON the standby system.
YES
Is the RESET/L.CLR switch of the standby system CPU in the neutral position (unreset)?
NO
Set the RESET/L.CLR switch of the standby system CPU from the RESET position to the neutral position (unreset).
YES
Is the tracking cable connected normally?
NO
Install the tracking cable normally.
YES
Is the standby system CPU normal?
NO
When the standby system CPU is in the error status, system switching is not executed.
YES
Is the mounted CPU a control system?
NO
When the mounted CPU is the standby system CPU, system switching is not executed.
YES
Has any reason for system switching failure (e.g. stop error of standby system CPU), if a system switching request is issued from the Ethernet module, occurred?
NO
Remove the reason for system switching failure not. Refer to the QnPRHCPU User's Manual (Redundant System) for the factor that does not switch the system.
YES
Is the CPU in the backup mode?
NO
Change the CPU from the separate mode to the backup mode.
YES
Please consult your local Mitsubishi service center or representative, explaining a detailed description of the problem.

(2) OPS connection communication error

The following shows a troubleshooting flowchart for the case where a communication error occurred during connection with the OPS.


Error in communication with OPS
Has "OPS connection" been set in the open system of the open setting?
NO
Set the open system of the open setting to "OPS connection". (Refer to Section 5.5.)
YES
Is the local station port number of the Ethernet module correct?
NO
Set the local station port number of the Ethernet module correctly. (Refer to Section 5.5.)
YES
Is the IP address of the OPS correct?
NO
Set the IP address of the OPS correctly.
YES
Is the MC protocol used?
YES
The MC protocol cannot be used for the connection set to OPS connection.
NO
Is the MELSOFT product used?
YES
MELSOFT connection cannot be made for the connection set to OPS connection.
NO
Is EZSocket used on the OPS side?
NO
The port set to OPS connection can be connected to only EZSocket.
YES
The possible cause is other than above or external device fault. Execute the troubleshooting for the case where data communication cannot be made. (Refer to Section 11.4.)

11 - 63

APPENDIX

Appendix 1 Function Upgrade for the Ethernet Module

The Ethernet module of function version B or later contains several functions not included in the earlier product (function version A).
This section gives a comparison of functions of the Ethernet module based on the function change or addition and explains program utilization and incorporation into an existing system.

Appendix 1.1 A comparison of functions of the Ethernet module

(1) A comparison of functions of the Ethernet module

The following shows the comparison of functions of the Ethernet module.

Function			QJ71E71-100	QJ71E71-B5	QJ71E71-B2
Data transmission rate	100 Mbps		○	×	×
	10 Mbps		○	○	○
Initial processing	Sequence program		○	○	○
	GX Developer network parameter setting		○	○	○
Re-initial processing	Sequence program		△	○	△
	Dedicated instruction		△	○	△
Open processing	Sequence program		○	○	○
	GX Developer network parameter setting		○	○	○
Communication using the MC protocol	QnA compatible 3E frame		○	○	○
	A compatible 1E frame		○	○	○
Communication using the fixed buffer	Procedure exist		○	○	○
	No procedure		○	○	○
Communication using the random access buffer			○	○	○
Sending/receiving e-mail	• Sending/receiving by PLC CPU • Sending by the PLC monitoring function (automatic notification function)		○	○	○
		Sending files in CSV format as attachment	○	○	△
		Sending main text	○	○	△
		Support for encoding/decoding	△	○	△
Communication using data link instructions			○	○	○
		Target station CPU type specification	△	△	△
File transfer (FTP server function)			○	○	○
Communication using the Web function			○	○	△
MELSECNET/H, MELSECNET/10 relay communication			○	○	○
Router relay communication (router relay function)			○	○	○
Existence check of remote devices (existence check function)	Checking by PING		○	○	○
	Checking by KeepAlive		△	○	△
Communication via pairing open			○	○	○
Communication via automatic open UDP port			○	○	○
Simultaneous broadcast			○	○	○
Support for the QCPU remote password function	Remote password check		○	○	△
	Unlock processing/ lock processing	Communication using the MC protocol	○	○	△
		GX Developer	○	○	△
		File transfer (FTP server) function	○	○	△
		Communication using the Web function	○	○	△

App - 1

App

Function			QJ71E71-100	QJ71E71-B5	QJ71E71-B2
Support for multiple CPU systems	Mounting to multiple CPU system stations		○	○	△
	Access to non-control PLC	Communication using the MC protocol	○	○	△
		GX Developer	○	○	△
		File transfer (FTP server) function	○	○	△
Mounting an Ethernet module to redundant system			△	△	△
Mounting an Ethernet module to the MELSECNET/H remote I/O station			○	○	△
Setting parameters to use Ethernet module functions			○	○	○
Accessing the QCPU via the Ethernet module (TCP/IP or UDP/IP)			○	○	○
Ethernet diagnostic function	Monitoring various Ethernet module statuses		○	○	○
	Via the Ethernet board	PING test	○	○	○
		Self-loop back test	○	○	△
	Via the CPU	PING test	○	○	△
Support for the IEEE802.3 frame			○	○	△
Connection of MELSOFT products (such as GX Developer)			○	○	○
	Simultaneous connection with multiple units via TCP/IP communication		○	○	△
	Simplifying access to other stations		△	○	△
	Access with the same station number		△	○	△
Hub connection status monitor function			△	×	×

○ : Can be used.

△ : Can be used. (However, there are restrictions on the product serial No. (manufactured date)) (∗1)

× : Cannot be used.

∗1 : There is a time limit on the manufacturing date of Ethernet modules that can support this function. For how to check the function version, see Section 2.7.

POINT

For the description of the following items, see Section 2.7.

- How to check the function version of the Ethernet module
- Correspondence with related products (CPU modules, GX Developer) that can use added functions.

App - 2

Appendix 1.2 Precautions when updating the module from function version A to function version B or later

This section explains the utilization of programs created for the Ethernet module of function version A and incorporation into an existing system.

(1) Program utilization

A program created for use with the Ethernet module of function version A can be used as is with the Ethernet module of function version B or later.

(2) Incorporation into an existing system

Wiring used for function version A can be used as is in the Ethernet module of function version B or later.

App - 3

Appendix 2 The QnA/A Series Module

This section gives a comparison of functions of the Ethernet module and QnA/A series module and explains program utilization and incorporation into an existing system.
The QnA/A series module refers to the following product:

Model name	Product name
AJ71E71	AJ71E71, AISJ71E71-B2, AISJ71E71-B5
AJ71E71-S3	AJ71E71-S3, AISJ71E71-B2-S3, AISJ71E71-B5-S3
AJ71E71N	AJ71E71N-T, AJ71E71N-B5, AJ71E71N-B2, AJ71E71N-B5T, A1SJ71E71N-T, AISJ71E71N-B5, AISJ71E71N-B2, A1SJ71E71N-B5T
QE71	AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2, A1SJ71QE71-B5
QE71N	AJ71QE71N-T, AJ71QE71N-B5, AJ71QE71N-B2, AJ71QE71N-B5T, A1SJ71QE71N-T, A1SJ71QE71N-B5, A1SJ71QE71N-B2, A1SJ71QE71N-B5T

Appendix 2.1 Functional comparisons between the Ethernet modules and QnA/A series modules

The following tables list the functional comparisons of the Ethernet modules and QnA/A series modules.
The ○ symbol in the model name columns indicates that the corresponding functions are compatible between the applicable models. (See each module's manual for details.)

indicates functions that have been added to or modified from the QnA/A series modules.

	Function		AJ71E71	AJ71E71-S3, AJ71E71N	QE71, QE71N		QJ71E71-100, QJ71E71-B5, QJ71E71-B2	Remark
					Products earlier than 9706	Products after 9706B		
1	Initial processing	Sequence program	○	○	○	○	○ (*3)	(*8)
		GX Developer parameter setting	×	×	×	○	○ (*3)	
2	Open processing	Sequence program	○	○	○	○	○ (*3)	(*9)
		GX Developer parameter setting	×	×	×	×	○ (*3)	
3	Communication using the fixed buffer	Procedure exist	○	○	○	○	○ (*4)	—
		No procedure	×	○	○	○	○ (*4)	—
4	Communication using the random access buffer		○	○	○	○	○ (*5)	—
5	Read/write of data in the PLC CPC (Communication using the MC protocol)		○	○	○	○	○	(*10)
6	Communication using data link instructions		×	×	×	○ (*2)	○	For communication between the PLC CPUs
7	Interrupt processing (at data receiving)	Fixed buffer communication	×	×	×	×	○	BUFRCVS instruction
		Data link instruction	×	×	×	×	○	RECVS instruction
8	Sending/receiving of e-mail	Sending/receiving with the sequence program	×	×	×	×	○	(*11)
		Sending with the automatic notification function	×	×	×	×	○	—
9	File transfer		×	×	×	○	○	FTP server function
10	Sending by the Web function		×	×	×	×	○	—
11	Simultaneous broadcast		×	○	○	○	○	Simultaneous broadcast function
12	Communication while the PLC CPU is in the STOP status		×	○	×	○	○ (*6)	—
13	Selection of communication data code (ASCII/binary)		○	○	○	○	○	—
14	MELSECNET/H, MELSECNET/10 relay communication		×	×	×	○ (*2)	○	—
15	Router relay function		×	○	○	○	○	Router relay function
16	Existence check of remote devices	Ping	×	○	○	○	○	—
		KeepAlive	×	×	×	×	○	—
17	Paring open communication		×	○	○	○	○	For communication using the fixed buffer
18	Unit of each timer setting value for data communication	500 ms	× (*1)	○	○ (Fixed)	○ (Fixed)	○ (Fixed)	—
		2 s	○ (Fixed)	○	×	×	×	—
19	Connection with GX Developer	TCP/IP	×	×	×	×	○	For 1:1 communication
		UDP/IP	×	○	○	○	○	Depends on the GX Developer function
20	Installation of EEPROM		×	×	○	○	× (*7)	Communication parameter registration
21	TCP Maximum Segment Size Option transmission		×	○ (*13)	×	○ (*13)	○ (*12)	—

○: The function is available ×: The function is unavailable

∗1 Applicable when the module's software version is earlier than "Q version".

∗2 The availability of the function depends on the date of manufacture/software version of the PLC CPU and SWnIVD/NX-GPPQ model GPP function software package.

∗3 A sequence program using input/output signals cannot be used together with the parameter settings from GX Developer.

∗4 Compatible with the input/output signals and the buffer memory of the QE71, QE71N.

∗5 The function is compatible, but cannot be used together with the e-mail sending/receiving function for sequence program.

∗6 When "Always wait for OPEN" is set in the network parameters of GX Developer, the conventional sequence programs are not required.
When using this function in the conventional sequence programs, the following conditions are not applicable. (It will not operate correctly because the same area is used.)
- "Always wait for OPEN" is set in the operational setting of the network parameters.
- "MELSOFT connection" is set in the open setting of the network parameters.
- Re-initial processing (with UINI instruction, buffer memory) is used in the sequence programs.

∗7 EEPROM is not installed. Items that were previously stored in the EEPROM in the QE71, QE71N are now stored by parameter setting from GX Developer.

∗8 By performing parameter setting from GX Developer, the initial processing is executed when the Ethernet module starts up. A conventional sequence program is not required.

∗9 The number of connections which can be opened from the PLC CPU has been increased to 16 for the Ethernet module. Also, when executing the Passive open processing for TCP/IP communication and the open processing for UDP/IP, the open processing is executed when the Ethernet module starts up by setting the "Always wait for OPEN" parameter from GX Developer. A conventional sequence program is not required.

∗10 When the Ethernet module is used, a maximum of 960 words can read/written (a maximum of 480 words for the QE71, QE71N).

∗11 This cannot be used together with the communication function using the random access buffer.

∗12 TCP Maximum Segment Size Option transmission is available for the product with the serial number which first 5 digits are 05051 or later.
- In the QJ71E71-100, QJ71E71-B5, QJ71E71-B2 with the first 5 digits of serial No. 05051 to 05081, the default value is preset to "Enable TCP Maximum Segment Size Option transmission". If normal communication cannot occur due to incorrect combination with the other node, change the setting to "Disable TCP Maximum Segment Size Option transmission".
- To change the TCP Maximum Segment Size Option transmission setting, change the TCP Maximum Segment Transmission setting area and execute re-initialization. (For re-initialization, refer to Section 5.2.3.)

∗13 TCP Maximum Segment Size Option transmission is applicable when the software version of the QE71N or AJ71E71N is "E" or later.
Note that the default is set to disable TCP Maximum Segment Size Option transmission. (Enabled when retransmitting TCP data)

POINT

The Ethernet module's response speed to an external device is faster than that of the Ethernet modules for the A/QnA series.
Keeping compatibility with the Ethernet modules for the A/QnA series strictly is not possible when using this Ethernet module. If this becomes a problem when considering the performance of an external device, try to make the timing similar to that of the conventional system using the QCPU constant scan setting or other applicable settings.

Appendix 2.2 Using the programs designed for QnA/A series modules

The data communication that was previously performed between the PLC CPU and an external device on the Ethernet using a QnA/A series Ethernet interface module (such as the AJ71E71) can also be performed using an Ethernet module.
The following explains how to use the programs designed for conventional modules to perform data communication using the Ethernet module.

(1) Using the programs designed for the AJ71E71(-S3) and AJ71E71N

(a) Using the programs designed of the external device side

The following parts of the communication function programs of the external device that are used for the AJ71E71(-S3) and AJ71E71N (hereinafter called the E71) can be utilized for communication with the Ethernet module. However, since the Ethernet module and the E71 have different response speeds, these programs may not be used as is. Make sure to test the operation first.

Program designed for E71 \ Destination		External device → Ethernet module	Ethernet module → External device	E71 → Ethernet module	Ethernet module → E71
Communication function	Communication using the fixed buffer (procedure exist)	○	○	○	○
	Communication using the random access buffer	○			
	Read/write data from/to the PLC CPU (*1)	○			

○: Communication can be performed by utilizing the programs for the E71 of the external device.

＊1 Only A compatible 1E frame commands can be used for data communication.
For details on the E71 commands, see the MELSEC Communication Protocol Reference Manual.
To perform data communication using other commands than A compatible 1E frame commands, create a new program.

(b) Using the sequence programs designed for the local station's E71.

The Ethernet module and the E71 have different buffer memory assignments, thus the sequence programs designed for the E71 cannot be utilized for the Ethernet module.
Create a new program referring to the chapters that explain each of the applicable functions.

(2) Using the programs designed for the AJ71QE71(N)

(a) Using the programs designed for the external device

The sequence programs of the external device that are designed for the AJ71QE71(N) (hereafter abbreviated as QE71) can be utilized for communication with the Ethernet module, except for the programs listed below:

- Programs for commands relating to file manipulation
  (Refer to the MELSEC Communication Protocol Reference Manual)
- Programs for accessing a data link system
  (QCPU (Q mode) modules cannot be connected to MELSECNET (II) and MELSECNET/B.)

However, since the Ethernet module and the QE71 have different response speeds, the programs may not be used as is. Make sure to test the operation first

(b) Using the programs designed for the local station's QCPU

1) Do not write the parameters of the Ethernet module set by GX Developer (network parameters) to the QCPU if sequence programs are used for initial processing or end processing.
If the parameter settings for the Ethernet module of GX Developer are not used, the following precautions should be observed when performing the communication:
- All the setting values of the communication condition setting switches for the QE71 will operate in OFF status. Set the communication conditions by the re-initial processing described in Section 5.2.3.
- MELSOFT products (e.g., GX Developer) cannot access the QCPU via the direct connection between MELSOFT products (e.g., GX Developer) and the Ethernet module.

2) If the parameter setting for the Ethernet module is performed using GX Developer, delete the sequence programs for initial processing and end processing.

3) The sequence programs designed for the local station's QE71 can be utilized for communication with the Ethernet module, except for the programs listed below:
- Sequence programs for accessing data link system.
(Refer to the QCPU User's Manual.)
- Sequence programs relating to EEPROM.
- Paring open settings for connection No. 8.
(See Section 5.7.1 "Paring Open".)
- Parameter setting program using the EPRSET instruction.

However, since the Ethernet module and the QE71 have different response speeds, the programs may not be used as is. Make sure to test the operation first.

REMARK

Keep the following points in mind when utilizing the programs.
- Do not use the sequence program for initial processing together with initial processing by parameter setting from GX Developer.
- Do not execute the following operations to the same connection simultaneously: open/close processing using input/output signals and open/close processing using the OPEN/CLOSE dedicated instructions, as well as the fixed buffer sending/receiving and sending/receiving processing using the BUFSND/BUFRCV/BUFRCVS instruction.

Furthermore, when setting parameters for the Ethernet module and communicating with the QCPU, make sure to use GX Developer (SW4D5C-GPPW-E or later).

POINT

(1) The operating mode and communication conditions can be set via the following parameter setting screen of GX Developer when using Q Series Ethernet modules:
- "Network Parameters Setting the number of MNET/10H Ethernet cards" screen
- "Ethernet operational setting" screen

(2) On Q Series Ethernet modules, there are no setting switches for making the operating mode settings and communication condition settings like there were on QnA/A Series Ethernet interface modules.

(3) The Q Series Ethernet module cannot cancel an open request before completing the open processing once the passive open processing is executed. Perform the close processing after the open processing is complete.

App - 8

Appendix 3 Installing the Ethernet Module on Existing Systems

The Ethernet module and the QnA/A series Ethernet interface module can coexist on the same Ethernet. The Ethernet module can be installed on the existing system's Ethernet using the existing wiring for the QnA/A series Ethernet interface module.

Appendix 4 Processing Time

Calculate the minimum processing time for each function using the expressions below. Note that the processing time may become longer depending on the load factor on the network (how congested the line is), the window size of each connected device, the number of connections used concurrently, and how the system is configured. Use the values obtained from the expressions below as a guideline for the processing time when communication is performed using only one connection.

(1) Minimum processing time of communication using the fixed buffer (communication between the Ethernet modules)

(a) Communication using the fixed buffer (Procedure exist)

$Tfs = St + Ke + (Kdf \times Df) + Sr$

Tfs : Time from the start of sending to the completion of sending (unit: ms)
St : Sending station scan time
Ke, Kdf : Constant (see the table below)
Df : Word count of send data
Sr : Receiving station scan time

	QJ71E71-100				QJ71E71-B5, QJ71E71-B2			
	Communication using TCP/IP		Communication using UDP/IP		Communication using TCP/IP		Communication using UDP/IP	
	Ke	Kdf	Ke	Kdf	Ke	Kdf	Ke	Kdf
Data communication using binary code	12	0.0065	10	0.0069	25	0.020	20	0.019
Data communication using ASCII code	12	0.030	10	0.029	26	0.068	21	0.068

(b) Communication using the fixed buffer (No procedure)

$Tfs = St + Ke + (Kdf \times Df)$

Tfs : Time from the start of sending to the completion of sending (unit: ms)
St : Sending station scan time
Ke, Kdf : Constant (see the table below)
Df : Byte count of send data

	QJ71E71-100				QJ71E71-B5, QJ71E71-B2			
	Communication using TCP/IP		Communication using UDP/IP		Communication using TCP/IP		Communication using UDP/IP	
	Ke	Kdf	Ke	Kdf	Ke	Kdf	Ke	Kdf
Data communication using binary code	7	0.0018	4	0.0014	16	0.0057	9	0.0025

[Calculation example]

Calculate the time from the start of sending to the completion of sending (unit: ms) when the QJ71E71-B5 communicate using TCP/IP and send 1017 words of binary code data using fixed buffer communication (procedure exist).

- Assume that the scan time on the receiving side is 10 ms and the scan time on the transmission side is 8 ms:

  $63.34 \text{ (ms)} \fallingdotseq 10 + 25 + (0.020 \times 1017) + 8$

(2) Minimum processing time of communication using the random access buffer

Trs = Kr + (Kdr × Df) + ACK processing time of an external device
(Added only for TCP/IP communication)

Trs : The time the Ethernet module takes to complete the processing of a data request from a PC after receiving it (unit: ms)
Kr, Kdr : Constants (see the table below)
Df : Word count of requested data
ACK processing time of the external device :
Time required until the external device returns an ACK after reading or writing of the random access buffer is completed.

		QJ71E71-100				QJ71E71-B5, QJ71E71-B2			
		Communication using TCP/IP		Communication using UDP/IP		Communication using TCP/IP		Communication using UDP/IP	
		Kr	Kdr	Kr	Kdr	Kr	Kdr	Kr	Kdr
Read	Data communication using binary code	3.1	0.004	2.1	0.005	9.4	0.008	6.6	0.008
	Data communication using ASCII code	3.1	0.016	2.2	0.016	9.1	0.030	6.5	0.030
Write	Data communication using binary code	3.1	0.006	2.1	0.005	9.5	0.014	6.6	0.012
	Data communication using ASCII code	3.1	0.017	2.2	0.015	9.6	0.042	6.7	0.036

[Calculation example 1]

Calculate the time the QJ71E71-B5 takes to complete the processing of a data request from a PC after receiving it, when the QJ71E71-B5 and the PC communicate using TCP/IP and read 508 words of binary code data from the random access buffer (unit: ms).

13.46 + ACK processing time of the external device (ms) ≒
9.4 (0.008 × 508) + ACK processing time of the external device

[Calculation example 2]

Calculate the time the QJ71E71-B5 takes to complete the processing of a data request from a PC after receiving it, when the QJ71E71-B5 and the PC communicate using TCP/IP and write 508 words of binary code data to the random access buffer (unit: ms).

16.61 + ACK processing time of the external device (ms) ≒
9.5 + (0.014 × 508) + ACK processing time of the external device

(3) Minimum processing time of communication using the MC protocol (batch read and batch write)

Tfs = Ke + (Kdt × Df) + Scr × number of scans required for processing + ACK processing time of external device

Tfs : Time from when Ethernet module receives request data from personal computer until it completes processing (unit: ms) ∗1

Ke, Kdt : Constant (refer to the table below)

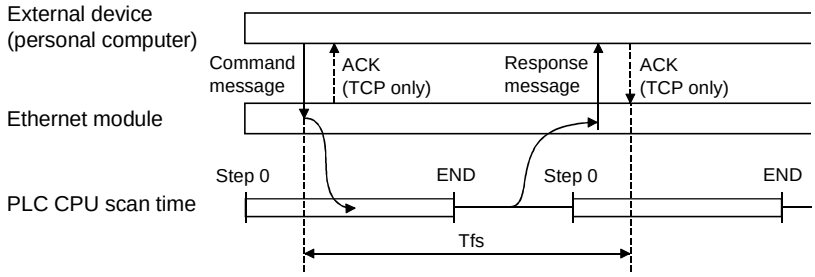
Df : Number of request data words + number of response data words (Application data part)

Scr : PLC CPU processing time

(a) When the target station is the QCPU
- Local station access:
  Ethernet module-mounted station scan time
- Other station access via MELSECNET/10
  Transmission delay time + Ethernet module-mounted station scan time

(b) When the target station is in the redundant system and data is relayed via the tracking cable
- Local station access:
  Control system CPU scan time + tracking time ∗2
- Other station access via MELSECNET/10
  Transmission delay time + control system CPU scan time + tracking time ∗2

∗1 The timing of the time from when the Ethernet module receives the request data from the personal computer until it completes processing is shown below.



∗2 When data is relayed via the tracking cable of the redundant system, add the tracking time. For the data transfer time in the tracking cable, refer to the QnPRHCPU User's Manual (Redundant System).

		QJ71E71-100				QJ71E71-B5, QJ71E71-B2			
		Communication using TCP/IP		Communication using UDP/IP		Communication using TCP/IP		Communication using UDP/IP	
		Ke	Kdt	Ke	Kdt	Ke	Kdt	Ke	Kdt
Batch read	Data communication using binary code	14	0.009	13	0.008	21	0.012	19	0.011
	Data communication using ASCII code	18	0.015	13	0.017	23	0.020	18	0.020
Batch write	Data communication using binary code	14	0.009	13	0.008	21	0.020	19	0.013
	Data communication using ASCII code	16	0.027	14	0.027	22	0.037	20	0.033

[Calculation example 1]

Calculate the time the QJ71E71-B5 takes to complete the processing of a data request from a PC after receiving it, when the QJ71E71-B5 and the PC perform TCP/IP communication and read 100 points of ASCII code data from the data register (D) of the local station (unit: ms) using the MC protocol communication.

- Assume that the scan time of the station in which QJ71E71-B5 is installed is 10 ms:

  37.64 + ACK processing time of the external device (ms) $\fallingdotseq$ 23 + (0.020 $\times$ (21 + 211)) + 10 $\times$ 1 + ACK processing time of the external device

  Command data length = 21 words
  Response data length = 211 words

[Calculation example 2]

Calculate the time the QJ71E71-B5 takes to complete the processing of a data request from a PC after receiving it, when the QJ71E71-B5 and the PC performs TCP/IP communication and write 100 points of ASCII code data to the data register (D) of the local station (unit: ms) using the MC protocol communication.

- When "Enable Write at RUN time" is set
- Assume that the scan time of the station in which QJ71E71-B5 is installed is 10 ms:

  40.58 (ms) $\fallingdotseq$ 22 + (0.037 $\times$ (221 + 11)) + 10 $\times$ 1

  Command data length = 221 words
  Response data length = 21 words

(4) Processing Time of the Dedicated Instructions

The following table shows approximate operation processing time of each dedicated instruction. The operation processing time differs slightly depending on the system configuration and communication protocol.

(a) QJ71E71-100

Instruction name	No. of access points		Processing time (unit: ms)						Instruction execution condition
			Q02CPU		Q02H/Q06H/Q12H/ Q25H/Q12PH/Q25PH/ Q12PRH/Q25PRHCPU		Q00J/Q00/Q01CPU		
	1)	2)	For 1)	For 2)	For 1)	For 2)	For 1)	For 2)	
BUFRCV	1 word	1017 words	1.3	1.8	0.9	1.4	2.2	5.8	TCP/IP communication, binary code communication, fixed buffer communication (procedure exist)
BUFRCVS			0.5	0.9	0.3	0.7	—	—	
BUFSND			12.8	19.2	11.5	18.1	14.0	23.5	
CLOSE	1 port		3.3		3.2		4.2		Closes the UDP/IP communication port
ERRCLR	Clear all the error information		2.2		2.0		3.4		—
ERRRD	Read initial abnormal code		1.2		0.8		2.4		—
OPEN	1 port		3.8		3.0		4.2		Opens the UDP/IP communication port
RECVS	1 word	480 words	0.6	0.8	0.3	0.5	—	—	Communication between stations in which Ethernet modules are installed
READ,SREAD			14.2	19.7	14.1 ∗1	18.7 ∗1	11.7	17.9	
RECV			2.1	3.2	2.0	2.9	1.8	4.3	
SEND			7.9	11.2	7.5	10.8	11.5	16.8	
WRITE,SWRITE			14.3	20.0	14.0 ∗1	19.2 ∗1	11.5	16.8	
ZNRD		230 words	14.4	17.1	13.8	16.6	12.1	14.1	
ZNWR			14.2	17.5	13.9	16.4	12.0	14.8	
UINI	—		21.5		21.2		21.8		Time from receiving the UINI instruction to the completion of re-initial processing (X19 is turned on)

∗1 When data is relayed via the tracking cable of the redundant system, add the tracking time. For the data transfer time in the tracking cable, refer to the QnPRHCPU User's Manual (Redundant System).

(b) QJ71E71-B5, QJ71E71-B2

Instruction name	No. of access points		Processing time (unit: ms)						Instruction execution condition
			Q02CPU		Q02H/Q06H/Q12H/ Q25H/Q12PH/Q25PH/ Q12PRH/Q25PRHCPU		Q00J/Q00/Q01CPU		
	1)	2)	For 1)	For 2)	For 1)	For 2)	For 1)	For 2)	
BUFRCV	1 word	1017 words	1.9	2.4	1.2	1.6	2.3	5.8	TCP/IP communication, binary code communication, fixed buffer communication (procedure exist)
BUFRCVS			0.5	0.9	0.3	0.7	—	—	
BUFSND			27.6	45.3	24.5	45.0	28.2	50.0	
CLOSE	1 port		4.5		4.5		6.0		Closes the UDP/IP communication port
ERRCLR	Clear all the error information		2.7		2.2		3.4		—
ERRRD	Read initial abnormal code		1.7		1.1		2.5		—
OPEN	1 port		4.3		3.3		5.2		Opens the UDP/IP communication port
RECVS	1 word	480 words	0.6	0.8	0.3	0.5	—	—	Communication between stations in which Ethernet modules are installed
READ,SREAD			30.1	41.1	27.7 ∗1	40.0 ∗1	27.7	38.9	
RECV			5.3	6.6	5.2	6.3	5.3	8.2	
SEND			21.4	30.4	20.3	29.1	22.8	30.8	
WRITE,SWRITE			30.0	41.8	29.4 ∗1	40.9 ∗1	28.2	37.9	
ZNRD		230 words	29.0	34.3	29.0	34.7	27.8	33.2	
ZNWR			29.7	36.4	29.4	35.2	27.6	33.4	
UINI	—		21.5		21.2		21.8		Time from receiving the UINI instruction to the completion of re-initial processing (X19 is turned on)

∗1 When data is relayed via the tracking cable of the redundant system, add the tracking time. For the data transfer time in the tracking cable, refer to the QnPRHCPU User's Manual (Redundant System).

(5) System switching time of redundant system

The following indicates the system switching time required when the Ethernet module mounted on the main base unit of the control system CPU in the redundant system issues a system switching request to the control system CPU at communication error or disconnection detection.
The system switching time is the time from when a communication error or disconnection is detected until the control system CPU is switched to the standby system CPU.

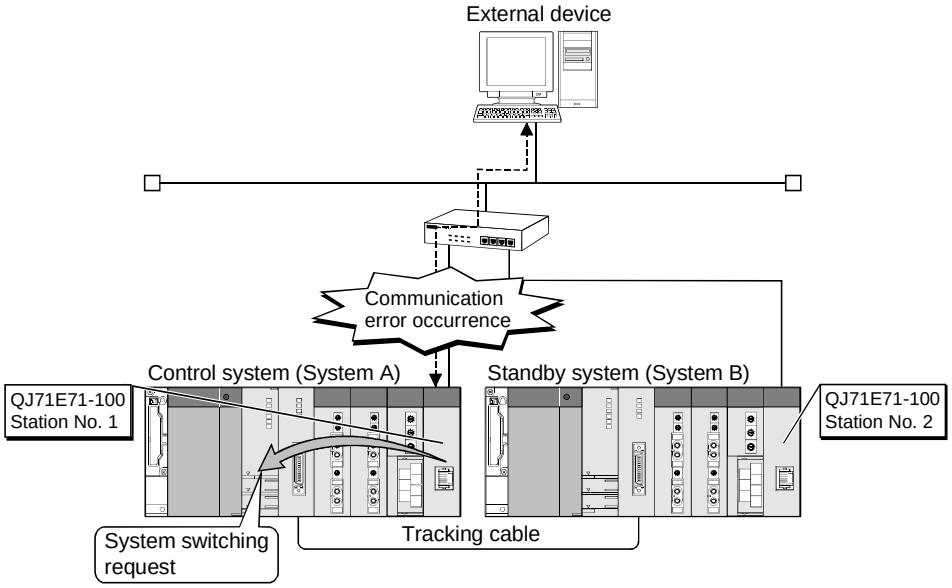
(a) When communication error is detected

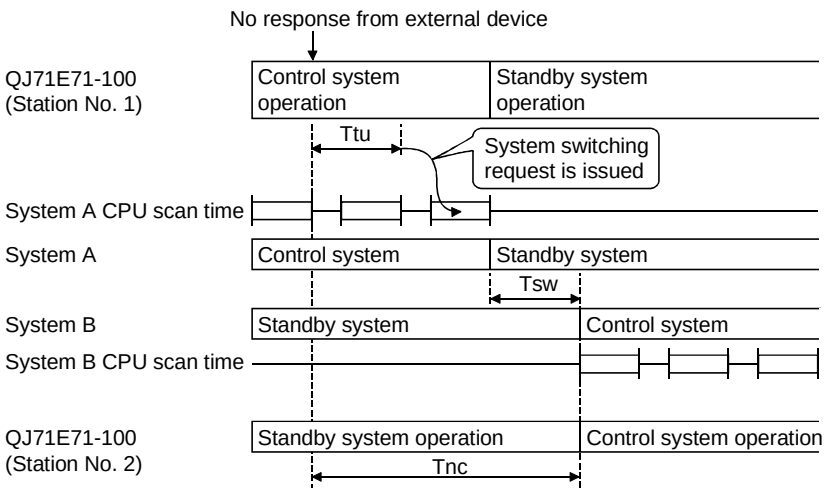
1) When ULP timeout occurs

Tnc = Ttu + St + Tcc

Tnc : System switching time
Ttu : TCP ULP timer value
St : 1 scan time
Tsw : CPU system switching time (Refer to the QnPRHCPU User's Manual (Redundant System).)

The following shows the system switching operation timing when a ULP timeout occurs.



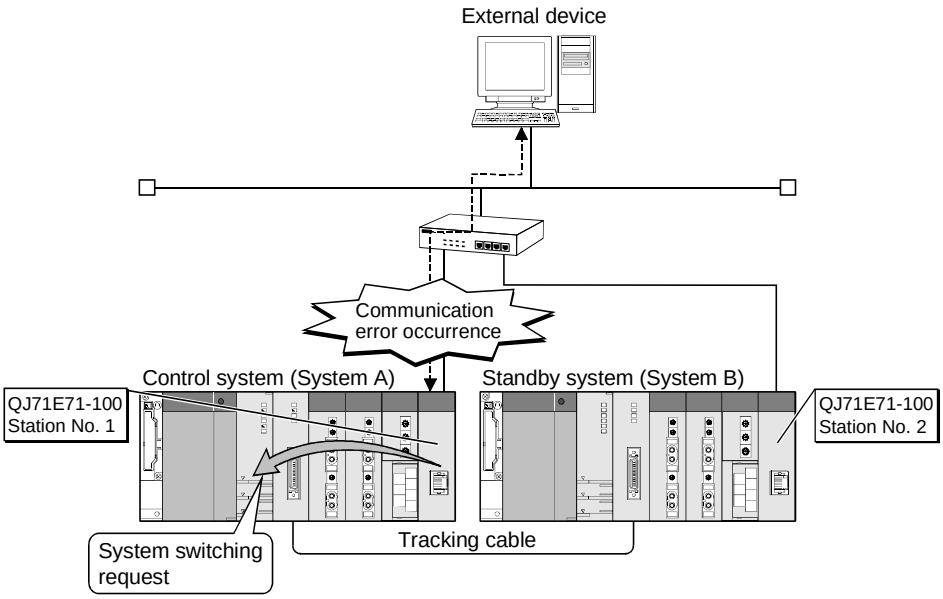


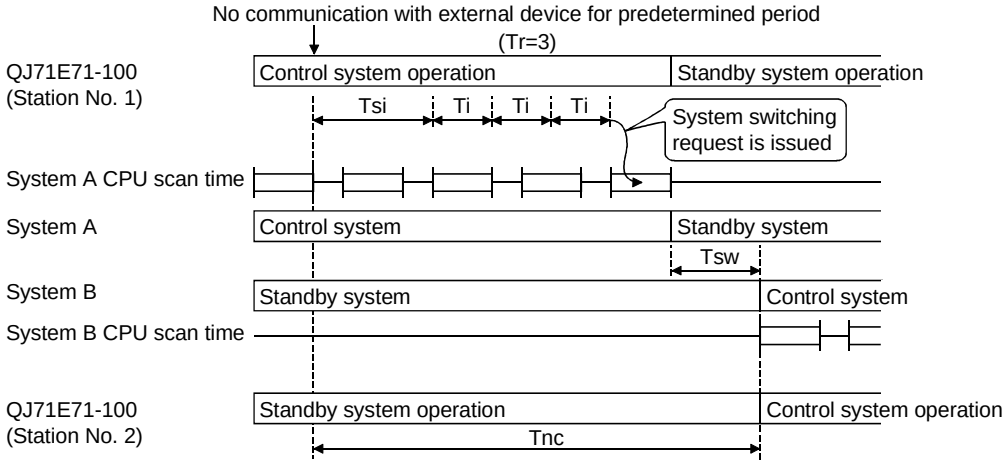
2) When existence check error occurs

Tnc = Tsi + Ti × Tr + St + Tcc

Tnc : System switching time
Tsi : Existence check starting interval timer value
Ti : Existence check timer value
Tr : Existence check resend count
St : 1 scan time
Tsw : CPU system switching time (Refer to the QnPRHCPU User's Manual (Redundant System).)

The following shows the system switching operation timing when an existence check error occurs.





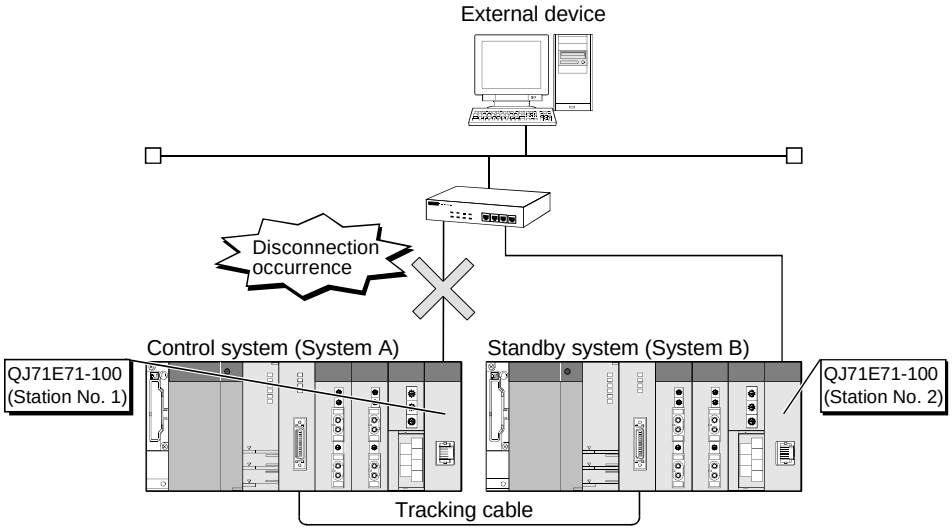
App - 15

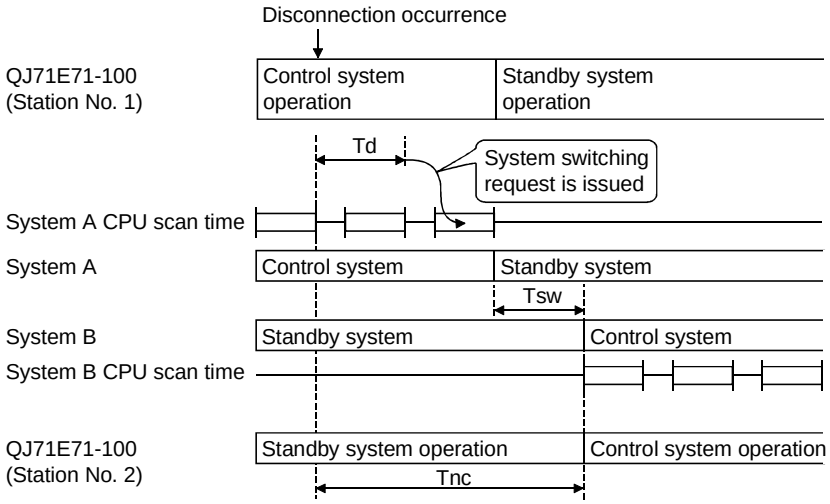
(b) When disconnection is detected

Tnc = Td + St + Tcc

Tnc : System switching time
Td : Cable disconnection timeout
St : 1 scan time
Tsw : System switching time (Refer to the QnPRHCPU User's Manual (Redundant System).)

The following shows the system switching operation timing when disconnection is detected.





App - 16

Appendix 5 ASCII Code List

LSD \ MSD		0	1	2	3	4	5	6	7
LSD		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	(SP)	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	–	=	M	]	m	}
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	_	o	DEL

Appendix 6 References

For details on TCP/IP, refer to the DDN Protocol Handbook (3 volumes).

Publisher

DDN Network Information Center
SRI International
333 Ravenswood Avenue, EJ291
Menlo Park, California 94025

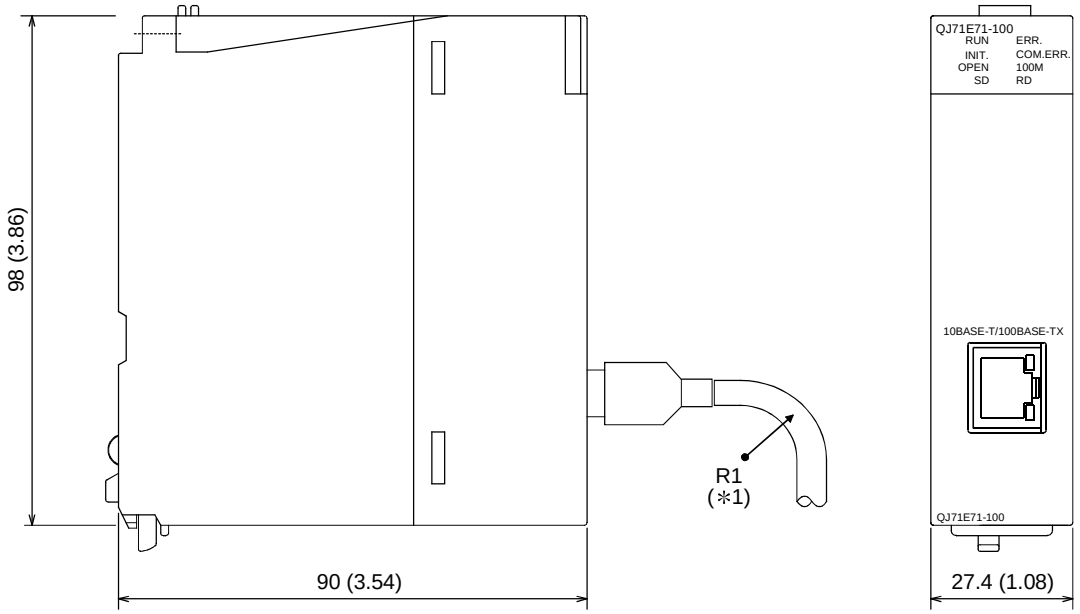
RFC Number

TCP	RFC793
UDP	RFC768
IP	RFC791
ICMP	RFC792
ARP	RFC826

App - 17

Appendix 7 External Dimensions

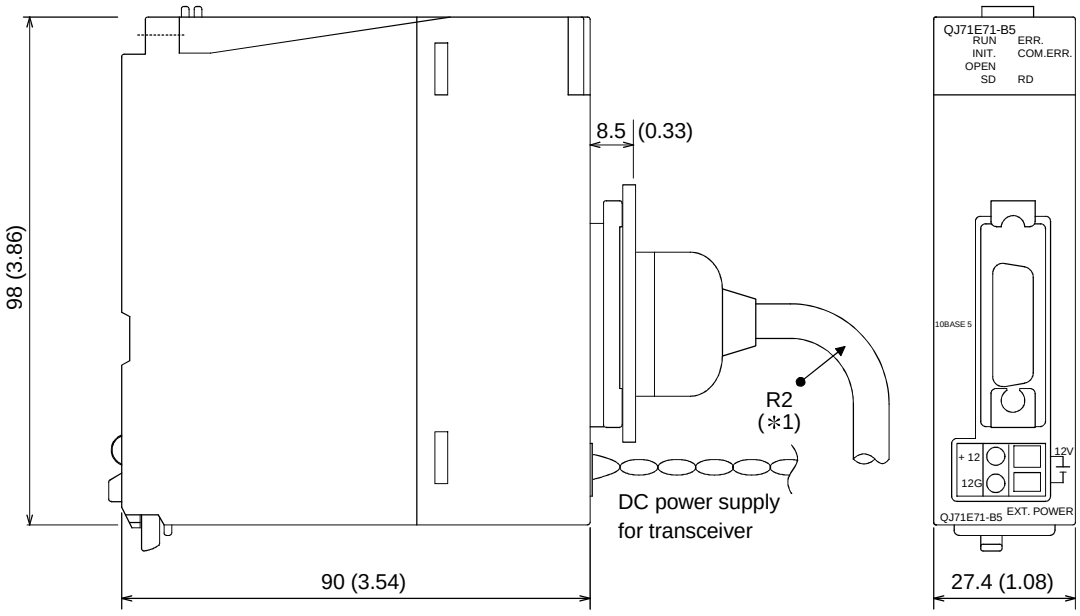
(1) QJ71E71-100



(Unit: mm (in.))

∗1 When connecting a twisted pair cable, set the bending radius near the connector (reference value: R1) as four times the cable's outside diameter or larger.

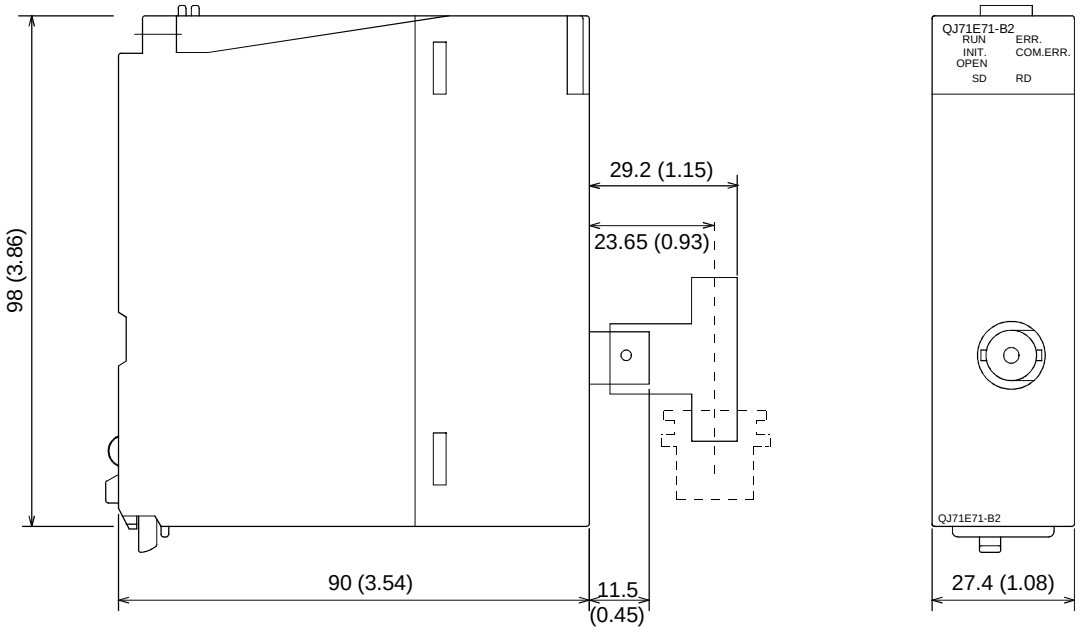
(2) QJ71E71-B5



(Unit: mm (in.))

∗1 When connecting an AUI cable, set the bending radius near the connector (reference value: R2) as four times the cable's outside diameter or larger.

(3) QJ71E71-B2



(Unit: mm (in.))

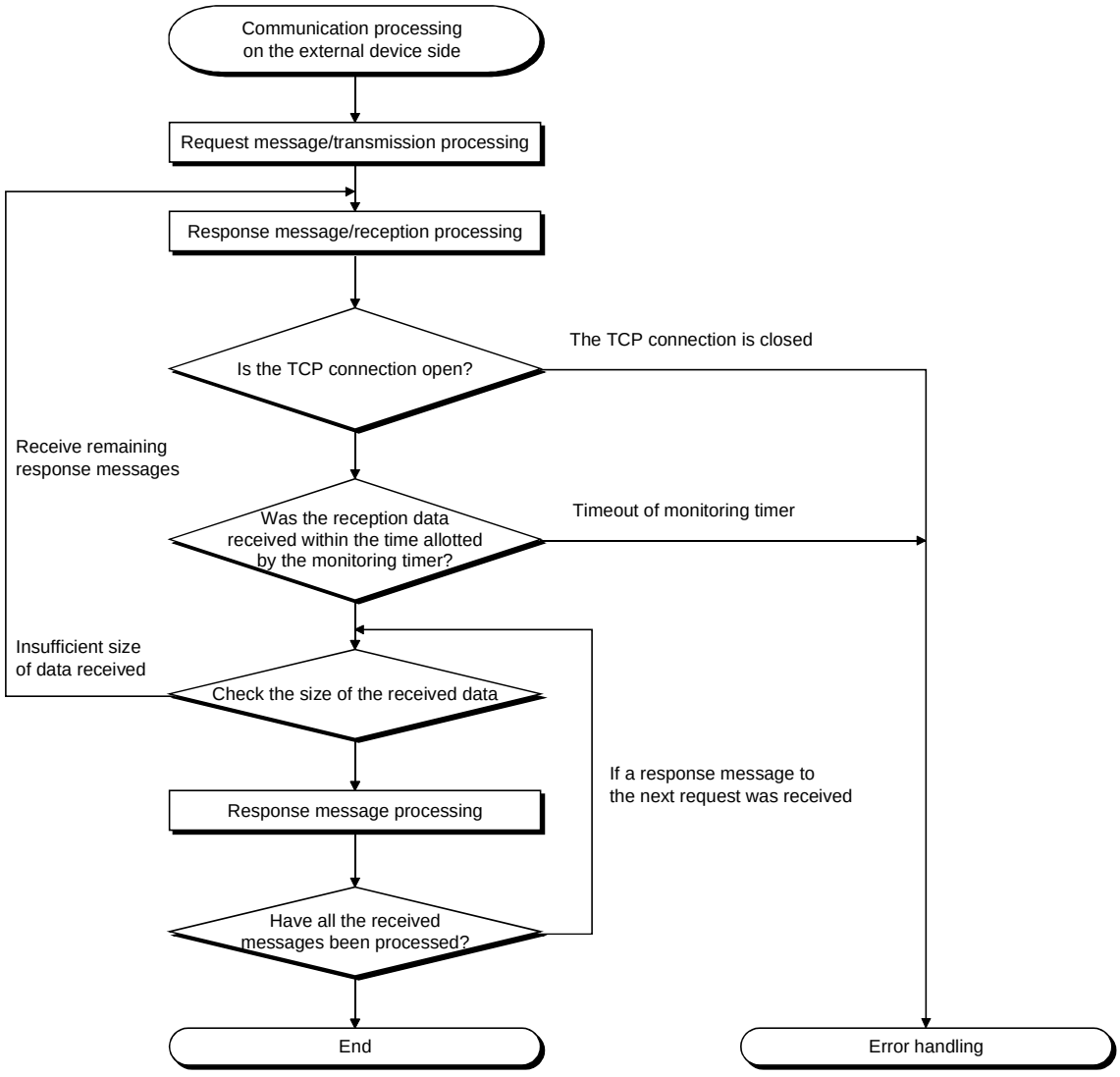
App - 19

Appendix 8 Program Examples

The program examples presented in this section test the connection between the Ethernet module and an external device (IBM PC/AT) that is connected to the same Ethernet.
For each program, only the minimum programming that is required to perform the communication test is done. Modify the IP address, port number and other values according to your system configuration. In addition, abnormal handling may be added.

(1) Receive processing of target device

A receive processing example on the target device side is shown below.


Communication processing on the external device side
Request message/transmission processing
Response message/reception processing
Is the TCP connection open?
The TCP connection is closed
Receive remaining response messages
Was the reception data received within the time allotted by the monitoring timer?
Timeout of monitoring timer
Insufficient size of data received
Check the size of the received data
If a response message to the next request was received
Response message processing
Have all the received messages been processed?
End
Error handling

Background

For Ethernet communications, the TCP socket functions are used inside the personal computer. However, these functions do not have any limits. Therefore, when the "send" function is executed once to transmit data, the receiving end (node) needs to execute the "recv" function once or more in order to read the data ("send" and "recv" is not proportional to 1:1 execution). For this reason, the receiving procedure explained above, is required.

(2) When the transmission target does not support the receive processing flow chart shown in section (1)

If the transmission target does not support the receive processing shown in Section (1), the following results will occur when communicating with the "TCP Maximum Segment Size Option transmission" is set to enable.

- Incorrect data read, when batch read is executed from the transmission target using MC protocol.
- Incorrect data read, after replacing the Ethernet module (which does not support the TCP Maximum Segment Size Option transmission function) with the alternative module supporting the function.
- Data is not received, even though the value of "Number of packet reception" area (Address: $1B8_H$, $1B9_H$) in the buffer memory was changed.

If this occurs, change the "TCP Maximum Segment Size Option transmission" setting to disable.

Appendix 8.1 Program example for communication using the MC protocol –1

The following explains a program, its execution environment and the contents of data communication .

(1) Execution environment of the program example

(a) PLC CPU side

1) QCPU model name of the Ethernet installed station : Q25HCPU
2) Ethernet module I/O signal : X/Y000 to X/Y01F
3) Ethernet module IP address : C0.00.01.FDH (192.00.01.253)
4) Ethernet module port number : 2000_H
5) GX Developer setting
   - Operational settings : See "(3) GX Developer setting (a)" on the next page
   - Open settings : See "(3) GX Developer setting (b)" on the next page

(b) External device side

1) Operation environment : Microsoft® Windows® 95 operating system
2) Ethernet interface board model name : WINSOCK compatible board
3) Library : WSOCK32.LIB
4) Software development environment : Microsoft® Corporation Visual C^{++} (Version 4.0)
5) Ethernet address : Setting not required because the ARP function is available
6) IP address : Receive at Active open
7) Port number : Receive at Active open

(c) Communication protocol : TCP/IP

(2) Outline of the program example

(a) Sequence program on the PLC CPU side
Parameters are set from GX Developer.
(Sequence program is not required)

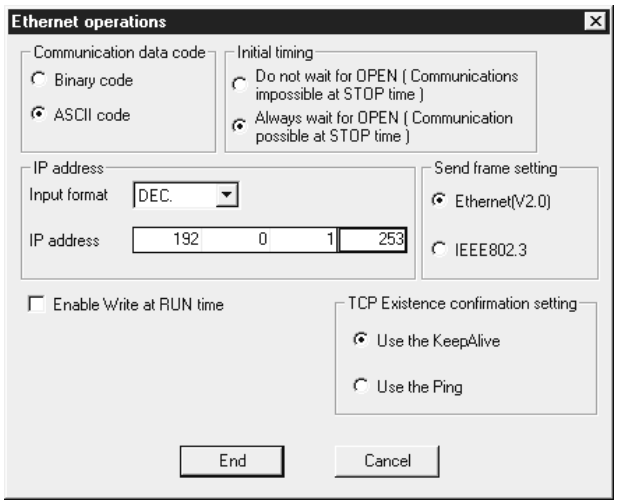
(b) Program on the external device side
Executes the following read/write data communication with the PLC CPU using the library mentioned above.

- Write in word units (for 5 points from D0 to D4)
- Read in word units (for 5 points from D0 to D4)

(3) GX Developer setting

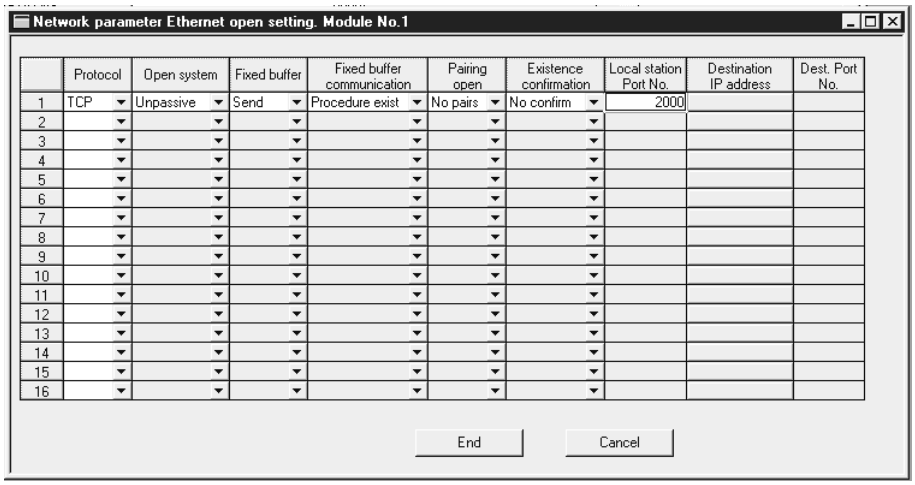
Set the PLC CPU parameters as follows.

(a) Operational settings



Local station IP address: C0.00.01.FDH (192.00.01.253)

(b) Open settings



Local station Port No.: 2000H

App - 22

(4) Program on the external device side

The program example of the external device shown below accesses the Q25HCPU of the station in which the Ethernet module is installed.
When this program is executed, the contents of the following communication messages are displayed in sequencer:

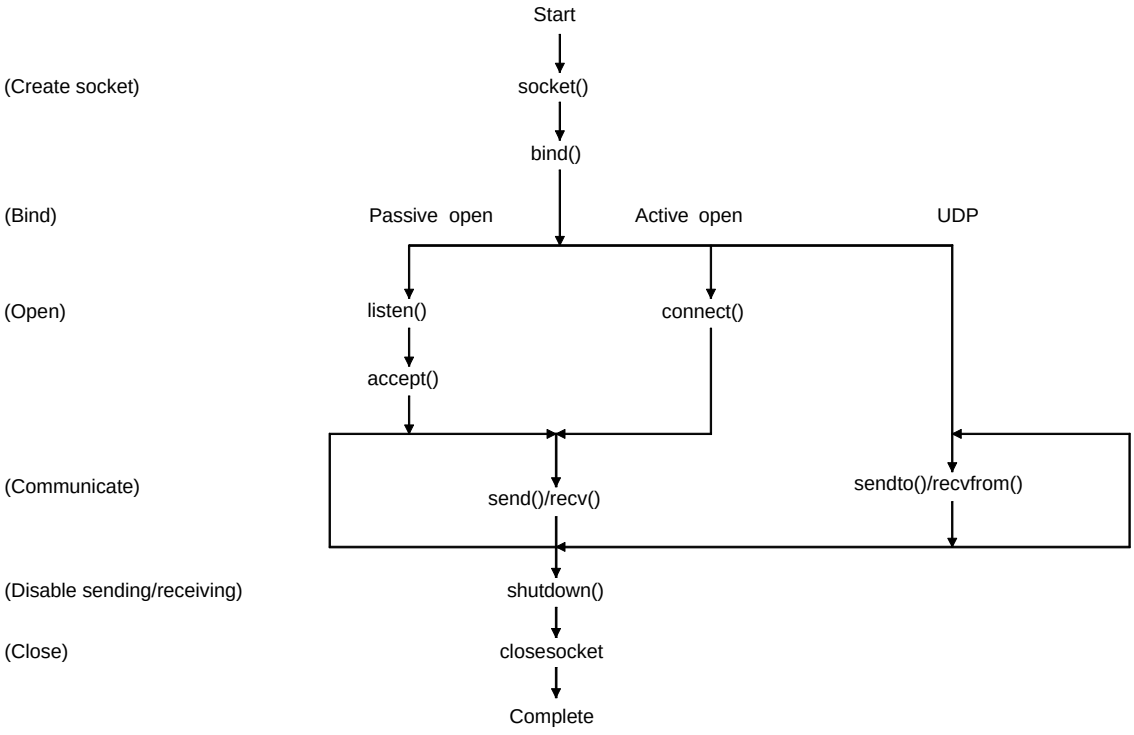
1) Batch write command message in word units
2) Batch write response message in word units
3) Batch read command message in word units
4) Batch read response message in word units

REMARK

(1) The following explains an outline of the compiling procedure for a program created using Microsoft® Corporation Visual C++ (Ver. 4.0)

1) Start Visual C++.
2) Prepare for program creation.
   Select File → New, and start the console application from the project workspace screen.
3) Open the file named AJSAMP.C and create a program.
   (See the program example on the next page.)
4) Compile the created file from the compile screen of the build menu.
5) Link WSOCK32.LIB from the setting screen of the build menu.
6) Create an executable file (AJSAMP.EXE) on the build screen of the build menu.
7) End Visual C++.
8) Execute AJSAMP.EXE.

(2) Outline of the procedure for calling the socket routine



App - 23

```
/ ********************************** /
/ **                                ** /
/ **    Sample program              ** /
/ **                                ** /
/ **    This program is a sample program to conduct a  ** /
/ **    connection test between the Ethernet module    ** /
/ **    and target device.          ** /
/ **    This program accesses the data register (D) of ** /
/ **    the PLC CPU installed together with the Ethernet ** /
/ **    module.                     ** /
/ **                                ** /
/ **    Copyright(C) 1996 Mitsubishi Electric ** /
/ **    Corporation                 ** /
/ **    All Rights Reserved         ** /
/ **                                ** /
/ ********************************** /


#include <stdio.h>
#include <winsock.h>

#define FLAG_OFF                0       // Completion flag OFF
#define FLAG_ON                 1       // Completion flag ON
#define SOCK_OK                 0       // Normal completion
#define SOCK_NG                -1       // Abnormal completion
#define BUF_SIZE             4096       // Receive buffer size

#define ERROR_INITIAL           0       // Initial error
#define ERROR_SOCKET            1       // Socket creation error
#define ERROR_BIND              2       // Bind error
#define ERROR_CONNECT           3       // Connection error
#define ERROR_SEND              4       // Send error
#define ERROR_RECEIVE           5       // Receive error
#define ERROR_SHUTDOWN          6       // Shutdown error
#define ERROR_CLOSE             7       // Line close error

//Definitions for checking the receiving sizes
//#define   RECV_ANS_1      4     // Receiving size of response message in reply to device write (1E frame)
#define RECV_ANS_1      22      // Receiving size of response message in reply to device write (3E frame)
//#define   RECV_ANS_2      24    // Receiving size of response message in reply to device read (1E frame)
#define RECV_ANS_2      42      // Receiving size of response message in reply to device read (3E frame)

typedef struct sck_inf{
        struct in_addr my_addr;
        unsigned short my_port;
        struct in_addr aj_addr;
        unsigned short aj_port;
};
```

```
int nErrorStatus;                                   // Error information storage variable
int Dmykeyin;                                       // Dummy key input
int Closeflag;                                      // Connection completion flag
int socketno;

int main()
{
        WORD wVersionRequested=MAKEWORD(1,1);       // Winsock Ver 1.1 request
        WSADATA wsaData;
        int length;                                 // Communication data length
        unsigned char s_buf[BUF_SIZE];              // Send buffer
        unsigned char r_buf[BUF_SIZE];              // Receive buffer
        int rbuf_idx;                               // Receive data storage head index
        int recv_size;                              // Number of receive data
        struct sck_inf sc;
        struct sockaddr_in hostdata;                // External device side data
        struct sockaddr_in aj71e71;                 // Ethernet module side data
        void Sockerror(int);                        // Error handling function

        unsigned long ulCmdArg ;                    // Non-blocking mode setting flag

        sc.my_addr.s_addr=htonl(INADDR_ANY);        // External device side IP address
        sc.my_port=htons(0);                        // External device side port number
        sc.aj_addr.s_addr=inet_addr("192.0.1.253"); // Ethernet module side IP address
                                                    // (C00001FDH)
        sc.aj_port=htons(0x2000);                   // Ethernet module side port number

        Closeflag=FLAG_OFF;                         // Connection completion flag off

        nErrorStatus=WSAStartup(wVersionRequested,&wsaData); // Winsock Initial processing

        if(nErrorStatus!=SOCK_OK) {
                Sockerror(ERROR_INITIAL);           // Error handling
                return(SOCK_NG);
        }

        printf("Winsock Version is %ld.%ld\n",HIBYTE(wsaData.wVersion),LOBYTE(wsaData.wVersion));
        printf("AJ_test Start\n");

        socketno=socket(AF_INET,SOCK_STREAM,0);     // Create socket for TCP/IP

        if(socketno==INVALID_SOCKET){
                Sockerror(ERROR_SOCKET);            // Error handling
                return(SOCK_NG);
        }

        hostdata.sin_family=AF_INET;
        hostdata.sin_port=sc.my_port;
        hostdata.sin_addr.s_addr=sc.my_addr.s_addr;
```

```
        if(bind(socketno,(LPSOCKADDR)&hostdata,sizeof(hostdata))!=SOCK_OK){
                                                        // Bind
                Sockerror(ERROR_BIND);                  // Error handling
                return(SOCK_NG);
        }

        aj71e71.sin_family=AF_INET;
        aj71e71.sin_port=sc.aj_port;
        aj71e71.sin_addr.s_addr=sc.aj_addr.s_addr;

        if(connect(socketno,(LPSOCKADDR)&aj71e71,sizeof(aj71e71))!=SOCK_OK){
                                                                // Connection (Active open)
                Sockerror(ERROR_CONNECT);                       // Error handling
                return(SOCK_NG);
        }

        Closeflag=FLAG_ON;                                      // Connection completion flag ON

        // Set to non-blocking mode
        ulCmdArg = 1;
        ioctlsocket(socketno, FIONBIO, &ulCmdArg);              // Set to non-blocking mode

//      strcpy(s_buf, "03FF000A44200000000005001122334455667788 99AA");
                                                        // D0 to D4 batch write request (1E frame)
        strcpy(s_buf, "500000FF03FF00002C000A14010000D*0000000005112233445566778899AA");
                                                        // D0 to D4 batch write request (3E frame)

        length=strlen(s_buf);

        if(send(socketno,s_buf,length,0)==SOCKET_ERROR){        // Data sending
                Sockerror(ERROR_SEND);                          // Error handling
                return (SOCK_NG);
        }
        printf("\n send data\n%s\n",s_buf);

        // Perform receiving size check and receiving processing simultaneously
        rbuf_idx = 0;                   // Receive data storage head index initialization
        recv_size = 0;                          // Initialize the number of receive data
        while(1) {
                length = recv(socketno, &r_buf[rbuf_idx], (BUF_SIZE - rbuf_idx), 0);
                                                                // Response data receiving
                if(length == 0) {                               // Is connection cut off?
                   Sockerror(ERROR_RECIEVE);                    // Error handling
                   return (SOCK_NG);
                }
```


App - 26

```
                if(length == SOCKET_ERROR) {
                    nErrorStatus = WSAGetLastError();
                    if(nErrorStatus != WSAEWOULDBLOCK) {
                        Sockerror(ERROR_RECIEVE);          // Error handling
                        return (SOCK_NG);
                    } else {
                        continue;                          // Repeat until messages are received
                    }
                } else {
                    rbuf_idx += length;                    // Update the receive data storage
                                                           // position
                    recv_size += length;                   // Update the number of receive data
                    if(recv_size >= RECV_ANS_1)            // Have all response messages been
                                                           // received?
                        break;                             // Stop repeating as messages have
                                                           // been received
                }
        }
        r_buf[rbuf_idx] = '\0' ;                           // Set NULL at the end of receive data

        printf("\n receive data\n%s\n",r_buf);

//      strcpy(s_buf, "01FF000A4420000000000500");         // D0 to D4 batch read request
                                                           // (1E frame)
        strcpy(s_buf, "500000FF03FF000018000A04010000D*0000000005");
                                                           // D0 to D4 batch read request
                                                           // (3E frame)

        length=strlen(s_buf);

        if(send(socketno,s_buf,length,0)==SOCKET_ERROR){   // Data sending
                Sockerror(ERROR_SEND);                     // Error handling
                return (SOCK_NG);
        }
        printf("\n send data\n%s\n",s_buf);

        // Perform receiving size check and receiving processing simultaneously
        rbuf_idx = 0;                                      // Receive data storage head index
                                                           // initialization
        recv_size = 0;                                     // Initialize the number of receive data
        while(1) {
                length = recv(socketno, &r_buf[rbuf_idx], (BUF_SIZE - rbuf_idx), 0);
                                                           // Response data receiving
                if(length == 0) {                          // Is connection cut off?
                    Sockerror(ERROR_RECIEVE);              // Error handling
                    return (SOCK_NG);
                }
```

```
        if(length == SOCKET_ERROR) {
                nErrorStatus = WSAGetLastError();
                if(nErrorStatus != WSAEWOULDBLOCK) {
                    Sockerror(ERROR_RECIEVE);                  // Error handling
                    return (SOCK_NG);
                } else {
                    continue;                                  // Repeat until messages are received
                }
        } else {
                rbuf_idx += length;                            // Update the receive data storage
                                                               // position
                recv_size += length;                           // Update the number of receive data
                if(recv_size >= RECV_ANS_2)                    // Have all response messages been
                                                               // received?
                    break;                                     // Stop repeating as messages have
                                                               // been received
        }
}
r_buf[rbuf_idx] = '\0' ;                                       // Set NULL at the end of receive data

printf("\receive data\n%s\n", r_buf);

        if(shutdown(socketno,2)!=SOCK_OK){                     // Processing to disable
                                                               // sending/receiving
                Sockerror(ERROR_SHUTDOWN);                     // Error handling
                return(SOCK_NG);
        }

        if(closesocket(socketno)!=SOCK_OK){                    // Close processing
                Sockerror(ERROR_CLOSE);                        // Error handling
                return(SOCK_NG);
        }

        Closeflag=FLAG_OFF;                                    // Connection completion flag off
        WSACleanup();                                          // Release Winsock.DLL

        printf("\nAJ_test End.\n\n Normally completed. \n");
        printf("Press any key to exit the program.\n");
        Dmykeyin=getchar();                                    // Wait for key input

        return(SOCK_OK);
}

void Sockerror(int error_kind)                                 // Error handling function
{
        if(error_kind==ERROR_INITIAL){
                printf("Initial processing is abnormal.");
        }
```

```
	else{
		nErrorStatus=WSAGetLastError();
		switch(error_kind){
		case ERROR_SOCKET:
			printf("Failed to create socket.");
			break;
		case ERROR_BIND:
			printf("Failed to bind.");
			break;
		case ERROR_CONNECT:
			printf("Failed to establish connection.");
			break;
		case ERROR_SEND:
			printf("Sending failed.");
			break;
		case ERROR_RECIEVE:
			printf("Receiving failed.");
			break;
		case ERROR_SHUTDOWN:
			printf("Failed to shutdown.");
			break;
		case ERROR_CLOSE:
			printf("Failed to close normally.");
			break;
		}
	}

	printf("Error code is %d.\n", nErrorStatus);

	if(Closeflag==FLAG_ON){
		nErrorStatus=shutdown(socketno,2);		// Shutdown processing
		nErrorStatus=closesocket(socketno);		// Close processing
		Closeflag=FLAG_OFF;				// Connection completion flag off
	}

	printf("Press any key to exit the program.\n");
	Dmykeyin=getchar();					// Wait for a key input
	WSACleanup();						// Release Winsock.DLL
	return;
}
```


App - 29

Appendix 8.2 Program example of communication using the MC Protocol -2

This section provides the program example, execution environment and data communication contents.

(1) Execution environment of program example

(a) PLC CPU side

1) Ethernet module-mounted station QCPU model name : Q25PRHCPU
2) I/O signals of Ethernet module : X/Y000 to X/Y01F
3) Ethernet module IP address (System A) : C0.00.01.FC$_H$ (192.00.01.252)
   Ethernet module IP address (System B) : C0.00.01.FD$_H$ (192.00.01.253)
4) Ethernet module port number : 2000$_H$
5) GX Developer settings
   - Operation setting : Refer to "(3) GX Developer settings (a)" on the next page.
   - Open setting : Refer to "(3) GX Developer settings (b)" on the next page.
   - Redundant settings : Refer to "(3) GX Developer settings (c)" on the next page.

(b) External device side

1) Operating environment :
   Microsoft® Windows® XP Professional Operating System
   Microsoft® Windows® XP Home Edition Operating System
2) Ethernet interface board model name : WINSOCK compatible board
3) Library : WSOCK32.LIB
4) Software development environment :
   Microsoft® Corporation make Visual C++ (Ver. 6.0) is used.
5) Ethernet address : Need not be set as the ARP function is available.
6) IP address : Received at Active open
7) Port number : Received at Active open

(c) Communication system : TCP/IP

(2) Overview of program example

(a) PLC CPU side sequence program

Parameter settings must be made using GX Developer.
(No sequence program required)

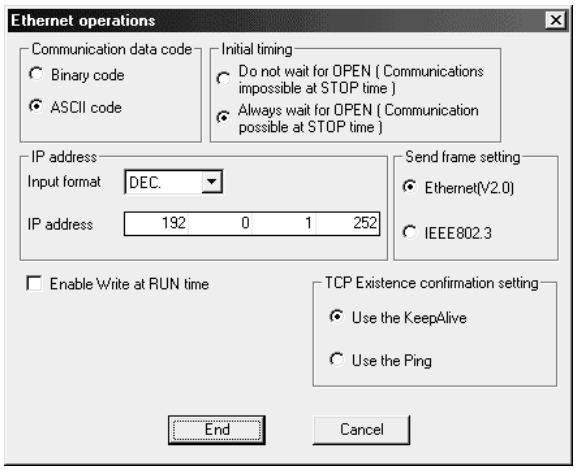
(b) External device side program

Writes data to the data registers D0 to D4 (5 points) of the redundant CPU (control system) in word unit using the above library.
If write of data from the system A side is unsuccessful due to a communication error or similar reason at this time, data are written from the system B side to the data registers of the redundant CPU (control system).

(3) GX Developer settings

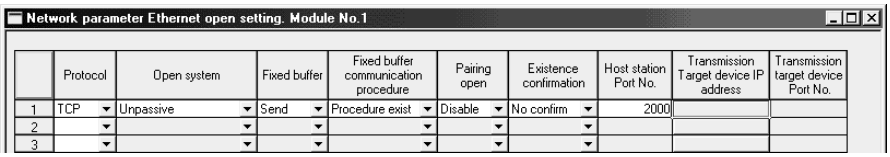
Set the parameters of the PLC CPU as described below.

(a) Operation setting



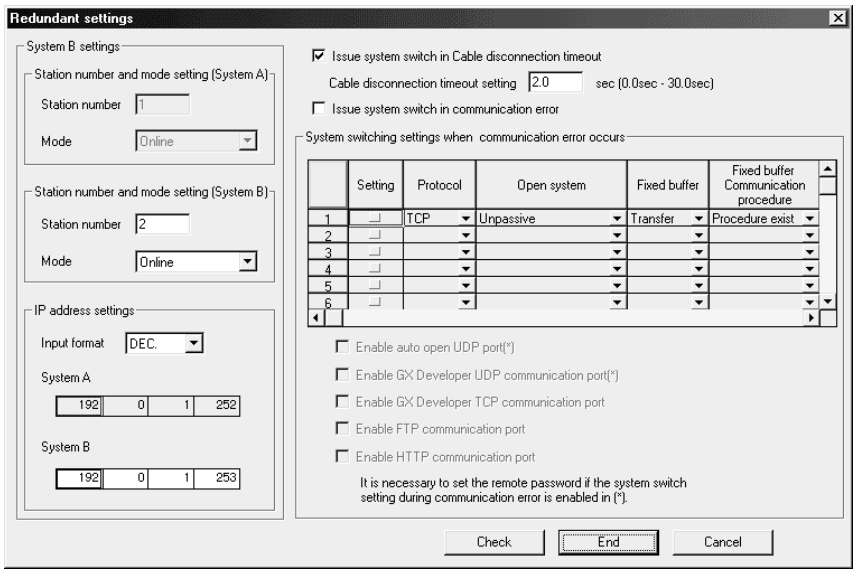
Ethernet module IP address (System A) : C0.00.01.FC$_H$ (192.00.01.252)

(b) Open setting



Local port number : 2000$_H$

(c) Redundant setting



Ethernet module IP address (System B) : C0.00.01.FD$_H$ (192.00.01.253)

(4) External device side program

The following shows a program example of the external device that accesses the Ethernet module-mounted station Q25PRHCPU.

When this program is executed, the following contents are displayed in due order.

1) Version of used Winsock
2) Test starting message
3) Write of batch command messages in word unit
4) Write of batch response messages in word unit
5) Test end message

REMARK

The following describes the general compile procedure for the program created using Microsoft® Visual C++ (Ver. 6.0).

1) Create a "QJSAMP.C" file using a text editor.
   (Refer to the next page for the program example.)
2) Start Visual C++.
3) Open the "QJSAMP.C" file.
   Choose File → Open, specify "QJSAMP.C" created in Step 1), and open it.
4) Execute the compile of the created program on the Compile sceen of the Build menu.
   Choose Build → Compile.
   Following the displayed messages, create a default project workspace.
5) Link WSOCK32.LIB from the Project setting screen.
   Choose Project → Setting to display the "Project setting" dialog box.
   Select the Link tab and add WSOCK32.LIB to the "Object/library module".
6) Create an execution file (QJSAMP.EXE) on the Build screen of the Build menu.
   Choose Build → Build.
7) Exit from Visual C++.
8) Execute QJSAMP.EXE.

App - 32

```
/ ********************************** /
/ **                                  ** /
/ **    Sample program                ** /
/ **                                  ** /
/ **    This program is a sample program for testing the  ** /
/ **    connection of the Ethernet module and external    ** /
/ **    device.                       ** /
/ **    This program accesses the data registers (D) of   ** /
/ **    the redundant CPU (control system) mounted        ** /
/ **    with the Ethernet module.     ** /
/ **                                  ** /
/ **    Copyright(C) 2004 Mitsubishi Electric Corporation ** /
/ **    All Rights Reserved           ** /
/ **                                  ** /
/ ********************************** /

#include <stdio.h>
#include <winsock.h>

#define FLAG_OFF            0                   // End flag OFF
#define FLAG_ON             1                   // End flag ON
#define SOCK_OK             0                   // Normal termination
#define SOCK_NG            -1                   // Unsuccessful termination
#define BUF_SIZE          4096                  // Receive buffer size

#define ERROR_NO_ERROR      0                   // No error
#define ERROR_INITIAL       1                   // Initial error
#define ERROR_SOCKET        2                   // Socket creation error
#define ERROR_BIND          3                   // Bind error
#define ERROR_CONNECT       4                   // Connect error
#define ERROR_SEND          5                   // Send error
#define ERROR_SHUTDOWN      6                   // Shutdown error
#define ERROR_CLOSE         7                   // Line close error

// Definition for checking receive size
#define RECV_ANS_1          22      // Response message receive size in reply to device write (3E frame)

typedef struct sck_inf{
        struct  in_addr my_addr;
        unsigned short  my_port;
        struct  in_addr qj_addr;
        unsigned short  qj_port;
};

int nErrorStatus;                               // Error information storage variable
int Dmykeyin;                                   // Dummy key input
int ShutdownflagA;                              // Shutdown flag (for System A connection)
int ShutdownflagB;                              // Shutdown flag (for System B connection)
```

```
int CloseflagA;                                    // Connection end flag (for System A connection)
int CloseflagB;                                    // Connection end flag (for System B connection)
int socketnoA;
int socketnoB;
int ConnectLastErrorA;                  // Connect processing error information (for System A connection)
int ConnectLastErrorB;                  // Connect processing error information (for System B connection)
int SendFlag;                           // Send completion flag


int main()
{
        WORD wVersionRequested = MAKEWORD(1, 1);                // Winsock Ver 1.1 request
        WSADATA wsaData;
        int length;                                             // Communication data length
        unsigned char s_buf[BUF_SIZE];                          // Send buffer
        unsigned char r_bufA[BUF_SIZE],r_bufB[BUF_SIZE];        // Receive buffer
        struct sck_inf scA,scB;
        struct sockaddr_in hostdataA,hostdataB;                 // External device side data
        struct sockaddr_in qj71e71A,qj71e71B;                   // Ethernet module side data
        BOOL DataRecv(int, unsigned char *, int);               // Receive processing function
        void Sockerror(int, int);                               // Error handling function

        unsigned long ulCmdArgA,ulCmdArgB;                      // Non-blocking mode setting flag

        scA.my_addr.s_addr = scB.my_addr.s_addr = htonl(INADDR_ANY);   // External device side IP address
        scA.my_port = scB.my_port = htons(0);                   // External device side port number
        scA.qj_addr.s_addr = inet_addr("192.0.1.252");
                                        // Ethernet module side IP address (System A: C00001FCh)
        scB.qj_addr.s_addr = inet_addr("192.0.1.253");
                                        // Ethernet module side IP address (System B: C00001FDh)
        scA.qj_port = scB.qj_port = htons(0x2000);              // Ethernet module side port number

        ShutdownflagA = ShutdownflagB = FLAG_OFF;               // Shutdown flag OFF
        CloseflagA = CloseflagB = FLAG_OFF;                     // Connection end flag OFF

        nErrorStatus = WSAStartup(wVersionRequested, &wsaData);     // Winsock initial processing

        ConnectLastErrorA = ERROR_NO_ERROR;
                                        // Connect processing error information initialization (for System A)
        ConnectLastErrorB = ERROR_NO_ERROR;
                                        // Connect processing error information initialization (for System B)

        if(nErrorStatus != SOCK_OK) {
                Sockerror(ERROR_INITIAL, ERROR_INITIAL);        // Error handling
                return (SOCK_NG);
        }

        printf("Winsock Version is %1d.%1d\n", HIBYTE(wsaData.wVersion), LOBYTE(wsaData.wVersion));
        printf("QJ_test Start\n");
```

```
// System A connect processing
socketnoA = socket(AF_INET, SOCK_STREAM, 0); // TCP/IP socket (for System A connection) creation
if(socketnoA != INVALID_SOCKET) {
        hostdataA.sin_family        = AF_INET;
        hostdataA.sin_port          = scA.my_port;
        hostdataA.sin_addr.s_addr = scA.my_addr.s_addr;

        if(bind(socketnoA, (LPSOCKADDR)&hostdataA, sizeof(hostdataA)) == SOCK_OK) {
                                                        // Bind (System A)
                qj71e71A.sin_family         = AF_INET;
                qj71e71A.sin_port           = scA.qj_port;
                qj71e71A.sin_addr.s_addr = scA.qj_addr.s_addr;

                if(connect(socketnoA, (LPSOCKADDR)&qj71e71A, sizeof(qj71e71A)) == SOCK_OK) {
                                                        // Connect (Active open: System A)
                        ShutdownflagA = FLAG_ON;        // Shutdown flag ON
                        CloseflagA = FLAG_ON;           // Connection end flag ON
                        // Set to non-blocking mode
                        ulCmdArgA = 1;
                        ioctlsocket(socketnoA, FIONBIO, &ulCmdArgA);
                                                // Set to non-blocking mode (for System A connection)
                } else {
                        ConnectLastErrorA = ERROR_CONNECT;    // Connection establishment failure
                }
        } else {
                ConnectLastErrorA = ERROR_BIND;         // Bind failure
        }
} else {
        ConnectLastErrorA = ERROR_SOCKET;               // Socket creation failure
}

// System B connect processing
socketnoB = socket(AF_INET, SOCK_STREAM, 0); // TCP/IP socket (for System B connection) creation
if(socketnoB != INVALID_SOCKET) {
        hostdataB.sin_family        = AF_INET;
        hostdataB.sin_port          = scB.my_port;
        hostdataB.sin_addr.s_addr = scB.my_addr.s_addr;

        if(bind(socketnoB, (LPSOCKADDR)&hostdataB, sizeof(hostdataB)) == SOCK_OK) {
                                                        // Bind (System A)
                qj71e71B.sin_family         = AF_INET;
                qj71e71B.sin_port           = scB.qj_port;
                qj71e71B.sin_addr.s_addr = scB.qj_addr.s_addr;
```

```
            if(connect(socketnoB, (LPSOCKADDR)&qj71e71B, sizeof(qj71e71B)) == SOCK_OK) {
                                                        // Connect (Active open: System B)
                ShutdownflagB = FLAG_ON;                // Shutdown flag ON
                CloseflagB = FLAG_ON;                   // Connection end flag ON
                // Set to non-blocking mode
                ulCmdArgB = 1;
                ioctlsocket(socketnoB, FIONBIO, &ulCmdArgB);
                                        // Set to non-blocking mode (for System B connection)
            } else {
                ConnectLastErrorB = ERROR_CONNECT;  // Connection establishment failure
            }
        } else {
            ConnectLastErrorB = ERROR_BIND;         // Bind failure
        }
    } else {
        ConnectLastErrorB = ERROR_SOCKET;           // Socket creation failure
    }

    // Connect completion processing
    if( (CloseflagA == FLAG_OFF) && (CloseflagB == FLAG_OFF) ){ // When both systems are abnormal
        Sockerror(ConnectLastErrorA, ConnectLastErrorB);        // Error handling
        return (SOCK_NG);
    }

    strcpy(s_buf, "500000FF03D000002C000A14010000D*00000000051122334455667788 99AA");
                                    // D0-D4 batch write request (3E frame, addressed to control system)
    length = strlen(s_buf);

    printf("Send starts. Press any key. \n");
    Dmykeyin = getchar();                               // Waiting for key input

    SendFlag = FLAG_OFF;                                // Send completion flag OFF
    // System A send processing
    if( CloseflagA == FLAG_ON && (SendFlag == FLAG_OFF) ){
        if(send(socketnoA, s_buf, length, 0) != SOCKET_ERROR) {     // Data send (System A)
            printf("\n Send data (System A) \n%s\n", s_buf);        // Send data display (System A)
            SendFlag = FLAG_ON;                                     // Send completion flag ON
            // Receive processing
            if(DataRecv(socketnoA, r_bufA, RECV_ANS_1) == TRUE) {   // Data receive
                printf("\n Receive data (System A) \n%s\n", r_bufA);    // Receive data display
            } else {
                printf("Receive failure (System A) \n");
            }
        } else {
            printf("Send failure (System A) \n");
        }
    }
```

```
// System B send processing
if( (CloseflagB == FLAG_ON) && (SendFlag == FLAG_OFF) ){
        if(send(socketnoB, s_buf, length, 0) != SOCKET_ERROR) {        // Data send (System B)
                printf("\n Send data (System B) \n%s\n", s_buf);       // Send data display (System B)
                SendFlag = FLAG_ON;                                    // Send completion flag ON
                // Receive processing
                if(DataRecv(socketnoB, r_bufB, RECV_ANS_1) == TRUE) {  // Data receive
                        printf("\n Receive data (System B) \n%s\n", r_bufB);   // Receive data display
                } else {
                        printf("Receive failure (System B) \n");
                }
        }else{
                printf("Send failure (System B) \n");
        }
}

// Send completion processing
if( SendFlag == FLAG_OFF ){
        Sockerror(ERROR_SEND, ERROR_SEND);          // Error handling
        return (SOCK_NG);
}

if(CloseflagA == FLAG_ON) {
        ShutdownflagA = FLAG_OFF;                   // Shutdown flag OFF
        if(shutdown(socketnoA, 2) != SOCK_OK) {     // Send/receive inhibit processing (System A)
                Sockerror(ERROR_SHUTDOWN, ERROR_NO_ERROR);     // Error handling
                return (SOCK_NG);
        }
}

if(CloseflagB == FLAG_ON) {
        ShutdownflagB = FLAG_OFF;                   // Shutdown flag OFF
        if(shutdown(socketnoB, 2) != SOCK_OK) {     // Send/receive inhibit processing (System B)
                Sockerror(ERROR_NO_ERROR, ERROR_SHUTDOWN);     // Error handling
                return (SOCK_NG);
        }
}

CloseflagA = FLAG_OFF;                              // Connection end flag OFF
if(closesocket(socketnoA) != SOCK_OK) {             // Close processing (System A)
        Sockerror(ERROR_CLOSE, ERROR_NO_ERROR);     // Error handling
        return (SOCK_NG);
}
```

```
	CloseflagB = FLAG_OFF;                                  // Connection end flag OFF
	if(closesocket(socketnoB) != SOCK_OK) {                 // Close processing (System B)
		Sockerror(ERROR_NO_ERROR, ERROR_CLOSE);         // Error handling
		return (SOCK_NG);
	}

	WSACleanup();                                           // Winsock.DLL release

	printf("\nQJ_test End. \n\nNormally completed. \n");
	printf("Program is closed. Press any key. \n");
	Dmykeyin = getchar();                                   // Waiting for key input

	return (SOCK_OK);
}


BOOL DataRecv(int socketno, unsigned char *pR_buf, int size_max)    // Receive processing function
{
	int length;                                             // Communication data length
	int rbuf_idx;                                           // Receive data storage starting index
	int recv_size;                                          // Number of received data

	// Performs receive processing while simultaneously making size check
	rbuf_idx = 0;                                   // Receive data storage starting index initialization
	recv_size = 0;                                  // Initializes the number of received data
	while(1) {
		length = recv(socketno, (pR_buf + rbuf_idx), (BUF_SIZE - rbuf_idx), 0);
                                                                // Response data receive
		if(length == 0) {                                       // Has connection been cut?
			return (FALSE);                                 // Error handling
		}
		if(length == SOCKET_ERROR) {
			nErrorStatus = WSAGetLastError();
			if(nErrorStatus != WSAEWOULDBLOCK) {
				return (FALSE);                         // Error handling
			} else {
				continue;                               // Repeated until data are received
			}
		} else {
			rbuf_idx += length;                     // Updates receive data storage position
			recv_size += length;                    // Updates the number of received data
			if(recv_size >= size_max)               // Have all response messages received?
				break;                          // Stops repeating as data are received
		}
	}
	*(pR_buf + rbuf_idx) = '\0' ;                           // At the end of received data
                                                                // set NULL
```

```
	return (TRUE);                                          // Normal termination
}


void Sockerror(int error_kind_A, int error_kind_B)          // Error handling function
{
	if (error_kind_A == ERROR_INITIAL){
		printf("Initial processing is abnormal. \n");
	}
	else{
		nErrorStatus = WSAGetLastError();
		switch(error_kind_A){
		case ERROR_SOCKET:
			printf("Socket could not be created. (System A)\n");
			break;
		case ERROR_BIND:
			printf("Bind could not be executed. (System A)\n");
			break;
		case ERROR_CONNECT:
			printf("Connection could not be established. (System A)\n");
			break;
		case ERROR_SEND:
			printf("Send could not be executed. \n");
			break;
		case ERROR_SHUTDOWN:
			printf("Shutdown could not be executed. (System A)\n");
			break;
		case ERROR_CLOSE:
			printf("Normal close could not be executed. (System A)\n");
			break;
		}
		switch(error_kind_B){
		case ERROR_SOCKET:
			printf("Socket could not be created. (System B)\n");
			break;
		case ERROR_BIND:
			printf("Bind could not be executed. (System B)\n");
			break;
		case ERROR_CONNECT:
			printf("Connection could not be established. (System B)\n");
			break;
		case ERROR_SHUTDOWN:
			printf("Shutdown could not be executed. (System B)\n");
			break;
		case ERROR_CLOSE:
			printf("Normal close could not be executed. (System B)\n");
			break;
```

```
            }
        }
        printf("Error code is %d. \n", nErrorStatus);

        if (ShutdownflagA == FLAG_ON){
            nErrorStatus = shutdown(socketnoA, 2);          // Shutdown processing (System A)
            ShutdownflagA = FLAG_OFF;                       // Shutdown flag OFF (System A)
        }
        if (ShutdownflagB == FLAG_ON){
            nErrorStatus = shutdown(socketnoB, 2);          // Shutdown processing (System B)
            ShutdownflagB = FLAG_OFF;                       // Shutdown flag OFF (System B)
        }

        if (CloseflagA == FLAG_ON){
            nErrorStatus = closesocket(socketnoA);          // Close processing (System A)
            CloseflagA = FLAG_OFF;                          // Connection end flag OFF (System A)
        }
        if (CloseflagB == FLAG_ON){
            nErrorStatus = closesocket(socketnoB);          // Close processing (System B)
            CloseflagB = FLAG_OFF;                          // Connection end flag OFF (System B)
        }

        printf("Program is closed. Press any key. \n");
        Dmykeyin = getchar();                               // Waiting for key input
        WSACleanup();                                       // Winsock.DLL release
        return;
}
```


App - 40

Appendix 8.3 Program example of communication using the MC Protocol –3

This section explains an example of an external device program that reads/writs data from/to the PLC CPU.
A sample program, its execution environment and contents of data communication are shown below.

(1) Execution environment of the program example

1) The settings of the PLC CPU side are the same as those of the execution environment described in Section 8.1 (1) (a) and (3) of Appendix.

2) The settings of the external device side are the same as those of the execution environment described in Section 8.1 (1) (b) of Appendix, except for the following including the software development:
   - Software development environment: Microsoft® Corporation Visual Basic (Ver. 6.0)
   - Arbitrary numbers are assigned for the IP address and port number.

3) The communication protocol is TCP/IP

(2) Outline of the program example

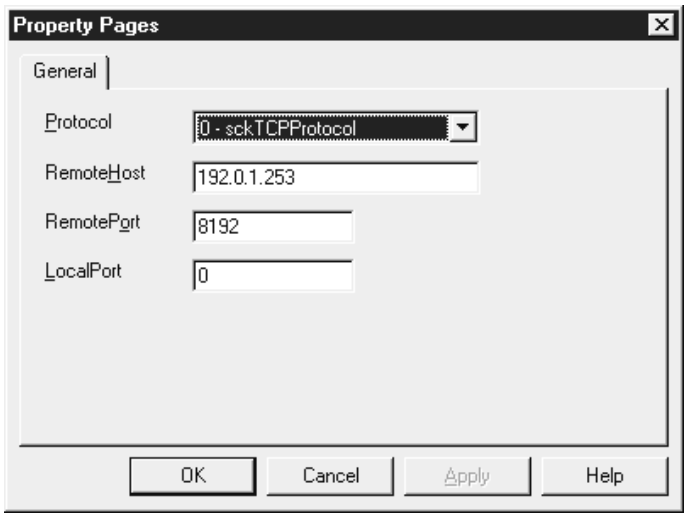
With the A-compatible 1E frame command (01: batch read in word units), this program reads data from D0 to D4 (five points) of the QCPU of the station on which the Ethernet module is mounted.

(3) Outline of the sample program

(a) Create a new project and form.

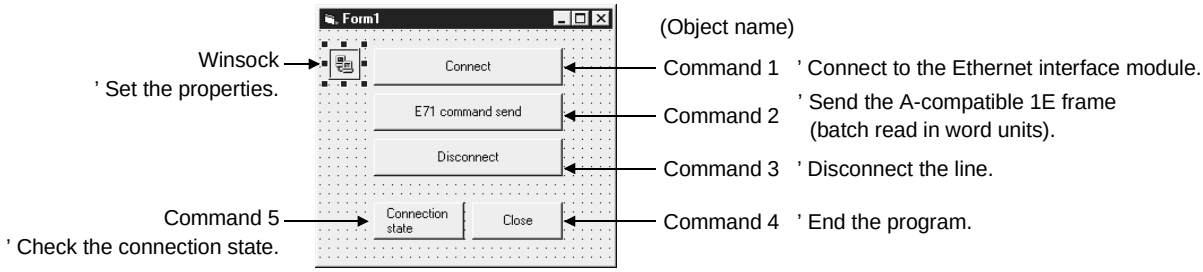
(b) Create the (example) window shown in (4) below using "Command Button" in the toolbox.

(c) Add "Microsoft Winsock Control 6.0" using the component's control.
Add a "Winsock" object from the toolbox to the form and set the Property Pages screen as follows:



(d) Create the program shown in (5).

(4) Window example (Form 1)



(5) Sample program (Form 1)

```
Option Explicit

Private Sub Command1_Click()
'Connect to the Ethernet interface module
        Me.Winsock1.Connect

End Sub

Private Sub Form_Load()
'When calling an Active open method from a PC to the Ethernet interface module,
'The property screen should be used or the settings performed as follows.
'Specify the protocol to be used.
'Winsock1.Protocol = sckTCPProtocol / sckUDPProtocol
'Specify the IP address of the Ethernet interface module.
'Winsock1.RemoteHost = "192.0.1.253"
'Specify the port No. used by the Ethernet interface module.
'Winsock1.RemotePort =8192              :H2000
'If the open method of the Ethernet interface module is Fullpassive open,
'specify the set port No.
'If the open method of the Ethernet interface module is Unpassive open,
'"0" - use any port No.
'Winsock1.LocalPort = 0                 :Unpassive open

End Sub

Private Sub winsock1_connect()
'Use the Connect event to perform confirmation processing at the time when the connection processing
is normally completed.
'The Connect event occurs when the connection processing is completed.
        MsgBox "Connection Completed"

End Sub
```

```
Private Sub Command2_Click()
Dim SData As String
'Read D0 to D4 (5 points) with the A-compatible 1E frame command.
        SData = "01ff000a4420000000000500"
'Send the data.
        Me.Winsock1.SendData SData

End Sub

Private Sub Command3_Click()
'Close the TCP connection socket (disconnect the line).
        Me.Winsock1.Close

End Sub

Private Sub Command4_Click()
'End the program.
        End

End Sub

Private Sub Command5_Click()
'Check the state of Winsock.
'See the Help of Visual Basic for details.
        MsgBox Winsock1.State

End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
'The DataArrival event occurs when new data arrives.

Dim RData As String
'Obtain the current data block and save it as a variant type variable.
'Read the response from the PLC CPU.
        Me.Winsock1.GetData RData
        MsgBox RData

End Sub
```


App - 43

Appendix 9 Communication Support Tool (MX Component)

MX Component is an ActiveX control library that supports any types of communication paths between IBM PC/AT compatible personal computers and PLCs. It allows the users to establish communication using simple processing only, without having to know about the different communication protocols used in the individual communication.
It also supports various programming languages, allowing for a wide range of application development.

This section provides the overview of the functions provided by MX Component and explains the procedure up to creation of applications.

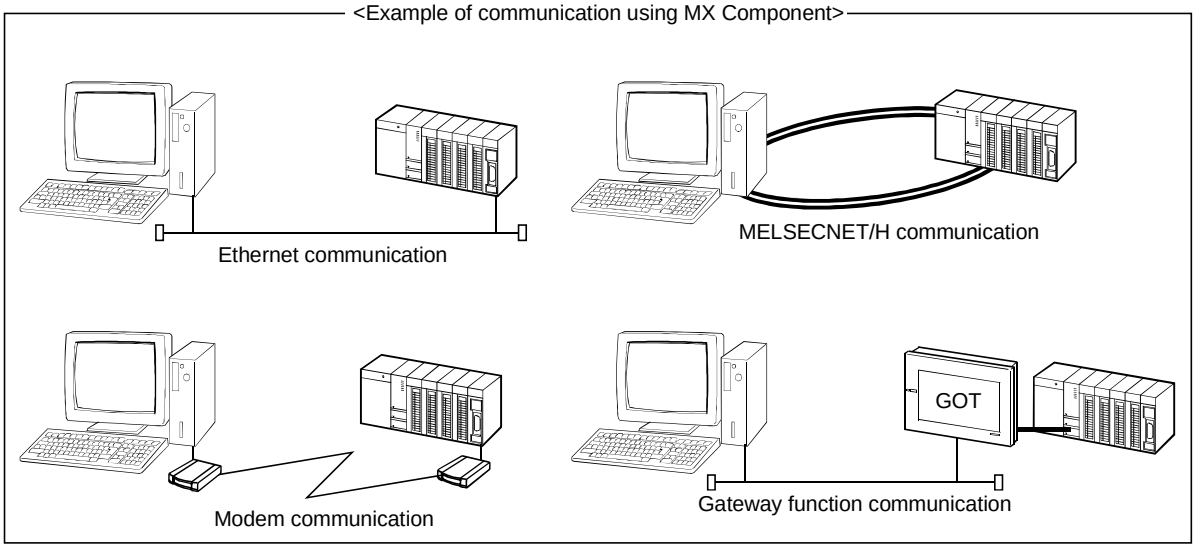
* Refer to Operating Manual and Programming Manual of MX Component for the details.

Appendix 9.1 Overview of MX Component

This section provides the overview of MX Component.
Different communication paths, operating systems, programming languages, and functions are supported depending on the version of MX Component used.

(1) Support for a wide range of communication paths to PLCs

MX Component supports a wide range of communication paths to PLCs. It is possible to construct systems according to the needs of the users.



(2) Dramatic improvement of application development efficiency

MX Component provides wizard-style communication setting utilities.
The user only needs to select settings from menus displayed on the screen in an interactive manner to achieve the communication settings required to access the target PLC CPU.
Moreover, once the communication setting has been performed, it is possible to access the PLC CPU simply by specifying the logical station number that is stored via the communication setting utilities.

App - 44

(3) Support for wide choice of basic operating systems

MX Component can run on IBM PC/AT compatible personal computers running the following basic operating systems:

- Microsoft® Windows® 95 Operating System
- Microsoft® Windows® 98 Operating System
- Microsoft® Windows NT® Workstation Operating System Version 4.0
- Microsoft® Windows® Millennium Edition Operating System
- Microsoft® Windows® 2000 Professional Operating System
- Microsoft® Windows® XP Professional Operating System
- Microsoft® Windows® XP Home Edition Operating System

(4) Support for a wide variety of programming languages

MX Component supports the following programming languages.
It allows the user to develop a wide range of customized applications.

Programming language	Development software
Visual Basic®	Microsoft® Visual Basic® 6.0
Visual C++®	Microsoft® Visual C++® 6.0
VBScript	Text editors and commercially available HTML tools
VBA	Microsoft® Excel 2000, Microsoft® Excel 2002, Microsoft® Access 2000 or Microsoft® Access 2002

(5) Support for functions dedicated for data communication with PLCs

MX Component provides the functions necessary for data communication with PLCs, including functions for opening/closing communication lines and reading/writing devices.
Multi-function communication programs can thus easily be developed with MX Component.

Function name	Function
Connect	Connects a telephone line.
Open	Opens a communication line.
Close	Closes a communication line.
Disconnect	Disconnects a telephone line.
GetErrorMessage	Displays error definition and corrective action.
ReadDeviceBlock	Batch-reads data from devices. (LONG type)
WriteDeviceBlock	Batch-writes data to devices. (LONG type)
ReadDeviceBlock2	Batch-reads data from devices. (SHORT type/INT type)
WriteDeviceBlock2	Batch-writes data to devices. (SHORT type/INT type)
ReadDeviceRandom	Randomly reads data from devices. (LONG type)
WriteDeviceRandom	Randomly writes data to devices. (LONG type)
ReadDeviceRandom2	Randomly reads data from devices. (SHORT type/INT type)
WriteDeviceRandom2	Randomly writes data to devices. (SHORT type/INT type)
SetDevice	Sets one device. (LONG type)
GetDevice	Acquires the data of one device. (LONG type)
SetDevice2	Sets one device. (SHORT type/INT type)
GetDevice2	Acquires data of one device. (SHORT type/INT type)

Function name	Function
ReadBuffer	Reads from buffer memory.
WriteBuffer	Writes to buffer memory.
GetClockData	Reads clock data from PLC CPU.
SetClockData	Writes clock data to PLC CPU.
GetCpuType	Reads a PLC CPU type.
SetCpuStatus	Remote RUN/STOP/PAUSE of PLC CPU
EntryDeviceStatus	Registers device status monitor.
FreeDeviceStatus	Deregisters device status monitor.
OnDeviceStatus	Announces event.

(6) Collecting data on Excel without programming

Using MX component and MX Sheet (SWnD5C-SHEET-E) allows users to collect PLC device data on Excel with only simple setting and without any programming.

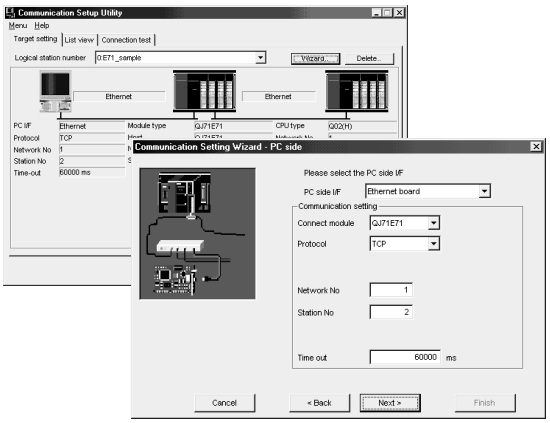
App - 46

Appendix 9.2 Usage Procedure of MX Component

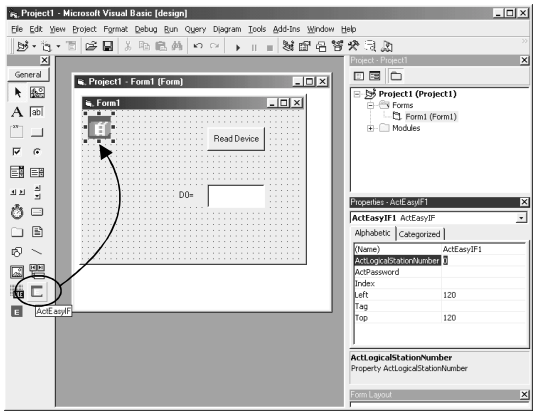
This section explains the procedure for creating programs and sample programs using MX Component.

(1) Procedure for creating programs

The procedure for creating programs is outlined below.
The usage procedure below uses Visual Basic® as an example.



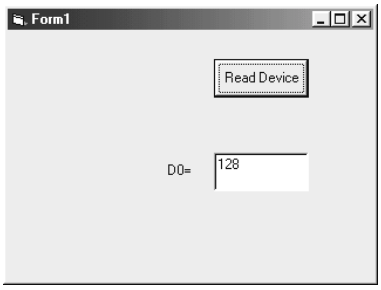
↓



↓

```
Command1                                   Click
Private Sub Command1_Click()

 ret = ActEasyIF1.GetDevice("D0", lpData(0))
 Label1.caption = lpData(0)

End Sub
```

↓



Completed

1) Perform the communication settings from a IBM PC/AT compatible personal computer to the PLC by following the wizard. (Some types of controls are set only by programs without using the wizard.)
The wizard allows the user to perform the settings required for the communication such as logical station number, connected module type, and PLC to be connected.

2) Paste the ACT control icon onto the form and assign the logical station number set in step 1 to the property of the pasted control.

3) Use the functions provided by the software to write a program that reads the device data.

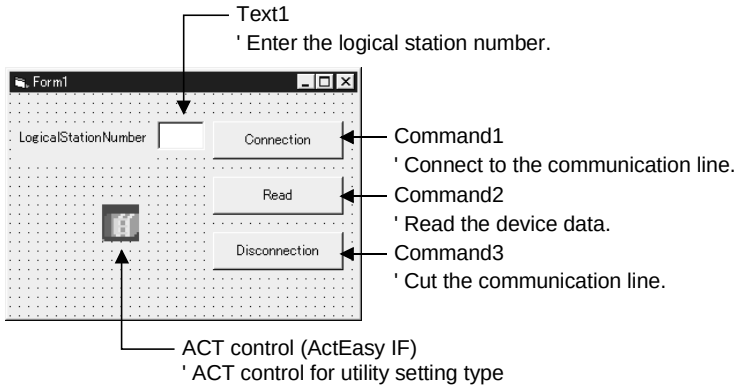
App - 47

(2) Sample program

The following sample program reads D0 to D4 (five points) of the target PLC using the logical station number.

(a) When Visual Basic® is used

1) Screen example (Form1)



2) Program example

```
Private Sub Command1_Click()
' ********************
'    Connection
' ********************
Dim rtn As Long
    ' Get LogicalstationNumber
    ActEasyIF1. ActLogicalStationNumber = Val(Text1.Text)
    ' Connection
    rtn = ActEasyIF1. Open()
    If rtn = 0 Then
        MsgBox "The connection was successful"
    Else
        MsgBox "Connection Error:" & Hex(rtn)
    End If
End Sub

Private Sub Command2_Click()
' ********************
'    Read
' ********************
Dim rtn As Long
Dim idata(5) As Integer
    ' D0-D4 are read
    rtn = ActEasyIF1. ReadDeviceBlock2 ("D0", 5, idata(0))
    If rtn = 0 Then
        MsgBox "D0-D5 = " & idata(0) & ", " & idata(1) & ", " & idata(2) & ", " & idata(3) & "," & idata(4)
    Else
        MsgBox "Read Error:" & Hex(rtn)
    End If
End Sub
```

```
Private Sub Command3_Click()
' ********************
'     Disconnection
' ********************
Dim rtn As Long

    ' Disconnection
    rtn = ActEasyIF1. Close()
    If rtn = 0 Then
        MsgBox "The disconnection was successful"
    Else
        MsgBox "Disconnection Error:" & Hex(rtn)
    End If
End Sub
```

(b) When Visual C++® is used

```
// ********************
//     Connection
// ********************
void CVCDlg::OnOpen()
{
        long lRet;
        CString szMessage;

        UpdateData();
    // Get LogicalStationNumber
        m_actEasyIF. SetActLogicalStationNumber ( m_lLogicalStationNumber ):
    // Connection
        lRet = m_actEasyIF. Open();
        if ( lRet == 0) {
                MessageBox ( "The connection was successful" )
        } else {
                szMessage. Format ( "Connection Error: %x", lRet );
                MessageBox ( szMessage )
        }
}
```

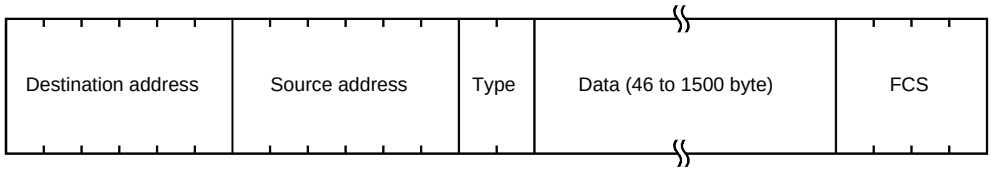
```
// ********************
//      Read
// ********************
void CVCDlg::OnRead()
{
        long lRet;
        short sData[5];
        CString szMessage;
      // D0-D4 are read
        lRet = m_actEasyIf. ReadDeviceBlock2 ( "D0", 5, sData );
        if ( lRet == 0) {
                szMessage. Format ( "D0-D5 = %d, %d, %d, %d, %d",
                                                sData[0], sData[1], sData[2], sData[3], sData[4] );
                MessageBox ( szMessage );
        } else {
                szMessage. Format ( "Read Error: %x", lRet );
                MessageBox ( szMessage )
        }
}

// ********************
//      Disconnection
// ********************
void CVCDlg::OnOpen()
{
        long lRet;
        CString szMessage;
        // Disconnection
          lRet = m_actEasyIF. Close();
          if ( lRet == 0) {
                    MessageBox ( "The disconnection was successful" )
        } else {
                    szMessage. Format ( "Disconnection Error: %x", lRet );
                    MessageBox ( szMessage )
        }
}
```

Appendix 10 Differences between the Ethernet and the IEEE802.3

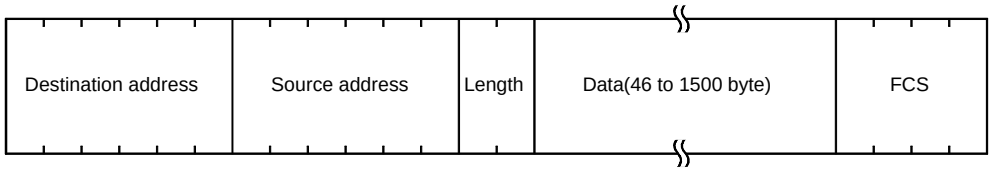
The following shows the Ethernet header in the data link layer supported by the Ethernet module.

Ethernet data link layer header	Ethernet module
Ethernet frame (V 2.0) specification	○
IEEE802.3 (ISO/IEC8802.3) frame specification	○

(1) Ethernet



(2) IEEE802.3



Appendix 11 ICMP Protocol Supported by the Ethernet Module

The following table outlines the types of ICMP supported by the Ethernet module and the processing performed by the Ethernet module.

ICMP Type	ICMP name/description	Processing by the Ethernet module
0	Echo Reply Result of IP packet loopback	Upon receiving an Echo Request, the Ethernet module sends this message.
3	Destination Unreachable Failed to deliver IP packet to the destination.	The Ethernet module sends this message when it receives data corresponding to a UDP connection that has not been opened.
8	Echo Request Requests loopback of IP packet	If destination existence confirmation is set in the buffer memory, the Ethernet module sends this message when attempting to confirm the existence of the target. (＊1)
Others	—	Ignored by the Ethernet module. (Not supported)

＊1 The Ethernet module can simultaneously receive two ICMP ECHO requests (type 8, Ping message), which are used for existence confirmation, etc., and handles them accordingly.
When three or more ICMP ECHO requests are received at the same time, the third and succeeding requests will be ignored.
If a response is not returned to the external device when an ICMP ECHO request is sent to the Ethernet module, send an ICMP ECHO request to the Ethernet module again.
The Ethernet module is able to receive a maximum of 1460-byte ICMP message at one time.
Do not send an ICMP message request exceeding 1460 bytes to the Ethernet module.

Appendix 12 Setting Value Recording Sheets

This section provides setting value recording sheets for parameters set with GX Developer. Make copies as needed.

Setting value recording sheet No.	GX Developer setting screen
Recording sheet 1	Network parameters Setting the number of MNET/10H Ethernet cards
	Ethernet operations
Recording sheet 2	Network parameters Ethernet initial settings
Recording sheet 3	Network parameters Ethernet open settings
Recording sheet 4	Network parameters Setting the Ethernet router relay parameter
Recording sheet 5	Network parameters Setting the Ethernet Station No. <-> IP information
Recording sheet 6	Network parameters Ethernet FTP parameter setting
Recording sheet 7	Network parameters Ethernet E-mail settings
	Ethernet mail address settings
Recording sheet 8	Ethernet News settings
Recording sheet 9	Network parameters Ethernet Interrupt settings
Recording sheet 10	Intelligent function module interrupt pointer setting
Recording sheet 11	Network parameters Setting the MNET/10 Ethernet routing information
Recording sheet 12	Remote password settings
Recording sheet 13	Network parameters Ethernet redundant settings

App - 52

Recording sheet 1

[Module number]

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Network parameters Setting the number of MNET/10H Ethernet cards	Starting I/O No.				
	Network No.				
	Group No.				
	Station No.				
	Mode			Online	
				Off line	
				Self refrain test	
				Hardware (H/W) test	
Ethernet Operations	Communication data code setting			Binary code communication	
				ASCII code communication	
	Initial Timing			Do not wait for OPEN (Communications impossible at STOP time)	
				Always wait for OPEN (Communication possible at STOP time)	
	IP address setting	Input format		Decimal	IP address Input format
				Hexadecimal	
		IP address	. . .		Adjust to input format
	Send frame setting			Ethernet (V2.0)	
				IEEE802.3	
	Enable Write at RUN time			Enable Write at RUN time	Enable: Check mark
	TCP Existence Confirmation setting			Use the KeepAlive	
				Use the Ping	

App - 53

Recording sheet 2

[Module number]

GX Developer setting screen	Data item		Setting data	
			Setting value	Remark
Network parameter Ethernet Initial settings	Timer setting	TCP ULP Timer		Default: 60 (500 ms)
		TCP zero window Timer		Default: 20 (500 ms)
		TCP resend timer		Default: 20 (500 ms)
		TCP end timer		Default: 40 (500 ms)
		IP assembly timer		Default: 10 (500 ms)
		Response monitoring timer		Default: 60 (500 ms)
		Destination existence confirmation starting interval		Default: 1200 (500 ms)
		Destination existence confirmation Interval timer		Default: 20 (500 ms)
		Destination existence confirmation resend timer		Default: 3 (times)
	DNS setting	Input format	Decimal / Hexadecimal	DNS server IP address Input format
		DNS server 1 IP address	. . .	Adjust to input format
		DNS server 2 IP address	. . .	
		DNS server 3 IP address	. . .	
		DNS server 4 IP address	. . .	

App - 54

Recording Sheet 3

[Module number]

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Network parameter Ethernet open settings	Connection No.	Protocol		TCP	
				UDP	
		Open system		Active	Setting not required if UDP is selected.
				Unpassive	
				Fullpassive	
		Fixed buffer		Send	
				Receive	
		Fixed buffer communication		Procedure exist	
				No procedure	
		Paring open		No pairs	
				Pairs	
		Existence confirmation		No confirm	
				Confirm	
		Local station Port No.			Input format: Hexadecimal
		Destination IP address		Decimal	Destination IP address input format
				Hexadecimal	
			. . .		Adjust to input format
		Destination Port No.			Input format: Hexadecimal
	Connection No.	Protocol		TCP	
				UDP	
		Open system		Active	
				Unpassive	
				Fullpassive	
		Fixed buffer		Send	
				Receive	
		Fixed buffer communication		Procedure exist	
				No procedure	
		Paring open		No pairs	
				Pairs	
		Existence confirmation		No confirm	
				Confirm	
		Local station Port No.			Input format: Hexadecimal
		Destination IP address		Decimal	Destination IP address input format
				Hexadecimal	
			. . .		Adjust to input format
		Destination Port No.			Input format: Hexadecimal

App - 55

Recording sheet 4

[Module number]

GX Developer setting screen	Data item		Setting data	
			Setting value	Remark
Network parameters Setting the Ethernet router relay parameter	Router relay function		☐ Do not use ☐ Use	
	Sub-net mask pattern		. . .	Adjust to input format
	Default router IP address		. . .	Adjust to input format
	Input format		☐ Decimal ☐ Hexadecimal	Router information Input format
	No.1	Sub-net address	. . .	Adjust to input format
		Router IP address	. . .	
	No.2	Sub-net address	. . .	
		Router IP address	. . .	
	No.3	Sub-net address	. . .	
		Router IP address	. . .	
	No.4	Sub-net address	. . .	
		Router IP address	. . .	
	No.5	Sub-net address	. . .	
		Router IP address	. . .	
	No.6	Sub-net address	. . .	
		Router IP address	. . .	
	No7	Sub-net address	. . .	
		Router IP address	. . .	
	No.8	Sub-net address	. . .	
		Router IP address	. . .	

App - 56

Recording sheet 5

[Module number]

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Network parameters Setting the Ethernet Station No. <-> IP information	Station No. <-> IP information			Automatic response system	
				IP address computation system	
				Table exchange system	
				Use-together system	
	Net mask pattern		. . .		Adjust to input format
	Input format			Decimal	Conversion setting input format
				Hexadecimal	
	No.	Network No.			Input format: Decimal
		Station No.			
		IP address	. . .		Adjust to input format
	No.	Network No.			Input format: Decimal
		Station No.			
		IP address	. . .		Adjust to input format
	No.	Network No.			Input format: Decimal
		Station No.			
		IP address	. . .		Adjust to input format
	No.	Network No.			Input format: Decimal
		Station No.			
		IP address	. . .		Adjust to input format
	No.	Network No.			Input format: Decimal
		Station No.			
		IP address	. . .		Adjust to input format

App - 57

Recording sheet 6

[Module number]

GX Developer setting screen	Data item		Setting data	
			Setting value	Remark
Network parameters Ethernet FTP parameter settings	FTP function settings		☐ Do not use ☐ Use	
	Login name			Default: "QJ71E71"
	Password	Current		
		New		
	Command input monitoring timer			Unit: Setting value × 500ms
	PLC CPU monitoring timer			

App - 58

Recording sheet 7

[Module number]

GX Developer setting screen	Data item			Setting data	
				Setting value	Remark
Network parameters Ethernet e-mail settings	General settings	Password			
		Mail address			
		Check of received mail		Check received mails	Check: Check mark
		Interval of inquiry			Set interval value to check received mail.
				s	Select unit for interval to check received mail.
				min	
				h	
	Mail Server name	Send setting	SMTP Server name		
			IP address	Decimal	IP address input format
				Hexadecimal	
				. . .	Adjust to input format
			POP Server name		
			IP address	Decimal	IP address input format
				Hexadecimal	
				. . .	Adjust to input format
Ethernet send mail address settings	No.	Send mail address			
		News setting		No execute	
				Execute news	
	No.	Send mail address			
		News setting		No execute	
				Execute news	
	No.	Send mail address			
		News setting		No execute	
				Execute news	
	No.	Send mail address			
		News setting		No execute	
				Execute news	
	No.	Send mail address			
		News setting		No execute	
				Execute news	
	No.	Send mail address			
		News setting		No execute	
				Execute news	

App - 59

Recording sheet 8

[Module number]

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Ethernet news settings	Condition for PLC inspection			No settings	
				Normal STOP	
				Module error/ module system error (Medium/Major level disorder)	
				Warning STOP (Minor disorder STOP)	
				Normal RUN	
				Warning RUN (Minor disorder RUN)	
				PAUSE	
	Send method			Send attached file	
				Send text mail	
	Attached file format			Binary	Adjust to input format
				ASCII	
				CSV	
	Attached file name				
	PLC inquiry Interval				Interval of CPU inquiry numeric value setting
				s	Interval of CPU inquiry unit selection
				min	
				h	
	Monitoring value input format			Decimal	Monitoring value input format
				Hexadecimal	
	No.	Condition device			
		Condition for inspection			
		Monitoring value			Adjust to input format
		News data storage device			
		News data			
	No.	Condition device			
		Condition for inspection			
		Monitoring value			Adjust to input format
		News data storage device			
		News data			
	No.	Condition device			
		Condition for inspection			
		Monitoring value			Adjust to input format
		News data storage device			
		News data			

App - 60

Recording sheet 9

[Module number]

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Network parameters Ethernet interrupt settings	Input format		☐	Decimal	Word device setting value input format
			☐	Hexadecimal	
	No.	Device code	☐	RECV instruction	
			☐	Fixed buffer	
		Device No.	—		Setting not required
		Detection method	Edge detection		Automatic setting
		Interrupt condition	Scan completed		Automatic setting
		Word device setting value	—		Setting not required
		Board No.			Input format: Decimal
		Interrupt (SI) No.			Input format: Decimal
	No.	Device code	☐	RECV instruction	
			☐	Fixed buffer	
		Device No.	—		Setting not required
		Detection method	Edge detect		Automatic setting
		Interrupt condition	Scan completed		Automatic setting
		Word device setting value	—		Setting not required
		Board No.			Input format: Decimal
		Interrupt (SI) No.			Input format: Decimal
	No.	Device code	☐	RECV instruction	
			☐	Fixed buffer	
		Device No.	–		Setting not required
		Detection method	Edge detect		Automatic setting
		Interrupt condition	Scan completed		Automatic setting
		Word device setting value	—		Setting not required
		Board No.			Input format: Decimal
		Interrupt (SI) No.			Input format: Decimal
	No.	Device code	☐	RECV instruction	
			☐	Fixed buffer	
		Device No.	—		Setting not required
		Detection method	Edge detect		Automatic setting
		Interrupt condition	Scan completed		Automatic setting
		Word device setting value	—		Setting not required
		Board No.			Input format: Decimal
		Interrupt (SI) No.			Input format: Decimal

App - 61

Recording sheet 10

GX Developer setting screen	Data item			Setting data	
				Setting value	Remark
Intelligent function module interrupt pointer setting	No.	PLC CPU side	Interrupt pointer start No.		Input format: Decimal
			Interrupt pointer No. of module		
		Intelligent module side	Start I/O No.		Input format: Hexadecimal
			Start SI No.		Input format: Decimal
	No.	PLC CPU side	Interrupt pointer start No.		
			Interrupt pointer No. of module		
		Intelligent module side	Start I/O No.		Input format: Hexadecimal
			Start SI No.		Input format: Decimal
	No.	PLC CPU side	Interrupt pointer start No.		
			Interrupt pointer No. of module		
		Intelligent module side	Start I/O No.		Input format: Hexadecimal
			Start SI No.		Input format: Decimal
	No.	PLC CPU side	Interrupt pointer start No.		
			Interrupt pointer No. of module		
		Intelligent module side	Start I/O No.		Input format: Hexadecimal
			Start SI No.		Input format: Decimal

App - 62

Recording sheet 11

GX Developer setting screen	Data item		Setting data	
			Setting value	Remark
Network parameters MNET/10H Ethernet routing information settings	No.	Transfer to network No.		Input format: Decimal
		Intermediate network No.		
		Intermediate station No.		
	No.	Transfer to network No.		
		Intermediate network No.		
		Intermediate station No.		
	No.	Transfer to network No.		
		Intermediate network No.		
		Intermediate station No.		
	No.	Transfer to network No.		
		Intermediate network No.		
		Intermediate station No.		
	No.	Transfer to network No.		
		Intermediate network No.		
		Intermediate station No.		
	No.	Transfer to network No.		
		Intermediate network No.		
		Intermediate station No.		
	No.	Transfer to network No.		
		Intermediate network No.		
		Intermediate station No.		

App - 63

Recording sheet 12

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Remote password settings	Password settings				
	Password active module settings	Model name	QJ71E71		
		Start X/Y			
Remote password detail settings	User connection No.			Connection 1	Select the object connection
				Connection 2	
				Connection 3	
				Connection 4	
				Connection 5	
				Connection 6	
				Connection 7	
				Connection 8	
				Connection 9	
				Connection 10	
				Connection 11	
				Connection 12	
				Connection 13	
				Connection 14	
				Connection 15	
				Connection 16	
	System connection			Auto open UDP port	
				FTP transmission port (TCP/IP)	
				GX Developer transmission port (TCP/IP)	
				GX Developer transmisstion port (UDP/IP)	
				HTTP port	

App - 64

Recording sheet 13

[Module number]

GX Developer setting screen	Data item		Setting data		
			Setting value		Remark
Redundant settings	Station number and mode setting (System B)	Station number			
		Mode		Online	
				Offline	
				Self-loopback test	
				H/W test	
	IP address settings	Input format		Decimal	IP address input format
				Hexadecimal	
		System B	. . .		Adjust to input format
	Issue system switch in cable disconnection timeout			Issue system switch in cable disconnection timeout	Issue: Check mark
	Cable disconnection timeout setting				Default: 2.0s
	Issue system switch in communication error			Issue system switch in communication error	Issue: Check mark
	System switching settings when communication error occurs			Connection 1	Select the object connection
				Connection 2	
				Connection 3	
				Connection 4	
				Connection 5	
				Connection 6	
				Connection 7	
				Connection 8	
				Connection 9	
				Connection 10	
				Connection 11	
				Connection 12	
				Connection 13	
				Connection 14	
				Connection 15	
				Connection 16	
				Enable auto open UDP port	
				Enable GX Developer UDP communication port	
				Enable GX Developer TCP communication port	
				Enable FTP communication port	
				Enable HTTP communication port	

App - 65

INDEX

[1 to 10]
10BASE2 ..... 2 - 7, 4 - 10
10BASE5 ..... 2 - 5, 4 - 9
10BASE-T ..... 2 - 4, 4 - 8
100BASE-TX ..... 2 - 3, 4 - 8

[A]
A compatible 1E frame ..... 6 - 2
(end codes) ..... 11 - 24
(abnormal codes) ..... 11 - 24
Abnormal code ..... 11 - 24
Accessible range
Fixed buffer communication (procedure exist) ..... 7 - 1
Fixed buffer communication (no procedure) ..... 8 - 1
Communication using the random access buffer ..... 9 - 1
Active open ..... 5 - 37, 5 - 42, 5 - 45
AJ71E71 ..... Appendix - 4
AJ71E71-S3 ..... Appendix - 4
Applicable systems ..... 2 - 1
Application data
Fixed buffer communication (procedure exist) ..... 7 - 13
Fixed buffer communication (no procedure) ..... 8 - 10
Communication using the random access buffer ..... 9 - 6
ARP ..... 1 - 15, 5 - 42
AUI cable (transceiver cable) ..... 2 - 5
Automatic open UDP port ..... 5 - 66

[B]
Buffer memory list ..... 3 - 13
Buffer memory ..... 3 - 12
BUFRCV instruction ..... 7 - 5, 8 - 6, 10 - 2
BUFRCVS instruction ..... 7 - 7, 8 - 8, 10 - 5
BUFSND instruction ..... 7 - 3, 8 - 4, 10 - 8

[C]
CLOSE instruction ..... 5 - 44, 10 - 11
Close processing ..... 5 - 44
Communication by e-mails ..... 1 - 4
Communication data code ..... 4 - 19
Communication method setting (TCP/UDP) ..... 5 - 37
Communication parameter ..... 3 – 14, 5 - 36
Communication procedure ..... 5 - 1
Communication status storage area ..... 3 - 15
Communication using the MC protocol ..... Chapter 6
Communication using the random access buffer ..... Chapter 9
Communication while the PLC CPU is in the STOP status ..... 4 - 20
Connection of connector for the coaxial cable ..... 4 - 11

[D]
Data communication using the MELSEC communication protocol ..... 1 - 2
Data format
Fixed buffer communication (procedure exist) ..... 7 - 12
Fixed buffer communication (no procedure) ..... 8 - 10
Communication using the random access buffer ..... 9 - 5
Data length
Fixed buffer communication (procedure exist) ..... 7 - 15
Communication using the random access buffer ..... 9 - 8
Dedicated instruction list ..... 3 - 8
Default router IP address ..... 5 - 18
Destination IP address ..... 5 - 40, 5 - 42
Destination Port No. ..... 5 - 41
DNS setting ..... 5 - 7
DNS ..... 1 - 15

[E]
Enable write at RUN time ..... 4 - 21
End code
Fixed buffer communication (procedure exist) ..... 7 - 16

Index - 1

Communication using the random access buffer ..... 9 - 10
End codes list ..... 11 - 22
ERRCLR instruction ..... 10 - 14, 11 - 4
Error code ..... 11 - 25
Error log area ..... 11 - 15
ERRRD instruction ..... 10 - 17, 11 - 4
Ethernet module ..... A - 21, 1 - 1
Ethernet ..... 1 - 1
Ethernet address ..... A - 21
Existence confirmation ..... 5 - 36

[F]
Fixed buffer (setting) ..... 5 - 38
Fixed buffer communication (no procedure) ..... Chapter 8
Fixed buffer communication (procedure exist) ..... Chapter 7
Fixed buffer communication (setting) ..... 5 - 38
FTP parameters ..... 4 - 14
FTP ..... 1 - 15
Full passive open ..... 5 - 37, 5 - 52
Functional comparisons between the QnA/A series modules ..... Appendix - 4

[G]
Generic terms and abbreviations ..... A - 21
GX Developer ..... A - 21
Setting item list ..... 3 - 9
Setting value recording sheets ... Appendix - 52
Group No. ..... 4 - 17

[H]
Head address ..... 9 - 8, 9 - 15
Header
Fixed buffer communication (procedure exist) ..... 7 - 12
Fixed buffer communication (no procedure) ..... 8 - 10
Communication using the random access buffer ..... 9 - 5

[I]
ICMP ..... 1 - 15, Appendix - 51
Improper access ..... 5 - 68
Initial processing ..... 5 - 3
Initial settings ..... 5 - 4
Initial timing settings ..... 4 - 20
Interrupt function
Fixed buffer communication (procedure exist) ..... 7 - 7
Fixed buffer communication (no procedure) ..... 8 - 8
Interrupt settings ..... 7 - 7
Interrupt pointer settings ..... 7 - 9
IP address setting
Destination IP address ..... 5 - 40
Local station IP address ..... 4 - 20
IP ..... 1 - 15

[L]
LED display ..... 4 - 6, 11 - 9
List of input/output signals ..... 3 - 10
Local station Port No. (setting) ..... 5 - 39
Local station (Ethernet module)
IP address ..... 4 - 20
Logical address ..... 9 - 15

[M]
MAC address ..... A - 21
MELSECNET/H, MELSECNET/10 relay communication ..... 3 - 6
Mode ..... 4 - 18
Monitoring function ..... 1 - 6
Multiple CPU system ..... 1 - 8, 2 - 8, 6 - 4
MX Component ..... 6 - 5, Appendix - 44

[N]
Network No. ..... 4 - 17
Network parameter
Number of MNET/10H Ethernet cards ... 4 - 16
Network type ..... 4 - 17

[O]
OPEN instruction ..... 5 - 43, 10-19
Open processing
Active open processing ..... 5 - 45
Passive open processing ..... 5 - 52
Pairing open processing ..... 5 - 63
Open settings ..... 5 - 36
Pairing open settings ..... 5 - 64
Sending by simultaneous broadcasting ..... 8 - 11

Index - 2

Receiving by simultaneous broadcasting ..... 8 - 13
Open system ..... 5 - 37
Operational settings ..... 4 - 19

[P]
Pairing open ..... 5 - 63
Parameter setting
Setting item list ..... 3 - 9
Setting screen list ..... 4 - 13
Setting value recording sheets... Appendix - 52
Passive open ..... 5 - 37, 5 - 42, 5 - 52
Physical address ..... 9 - 15
POP3 ..... 1 - 15
Port No. ..... 5 - 40, 5 - 41
Program example
Open/close processing (Active) ..... 5 - 49
Open/close processing (Passive) ..... 5 - 57
Fixed buffer communication (procedure exist) ..... 7 - 18
Fixed buffer communication (no procedure) ..... 8 - 18
Protocol ..... 5 - 37

[Q]
QE71 ..... Appendix - 4
QnA compatible 3E frames ..... 6 - 2

[R]
Re-initial processing ..... 5 - 9
Redundant System ..... 1 - 11, 5 - 89
Related manuals ..... A - 13
Remote password check ..... 5 - 68
Remote password mismatch notification accumulated count ..... 5 - 85
Response format
Fixed buffer communication (procedure exist) ..... 7 - 13
Communication using the random access buffer ..... 9 - 6
Router IP address ..... 5 - 20
Router relay function ..... 5 - 17
Router relay parameter ..... 5 - 16

[S]
Setting the number of cards ..... 4 - 13, 4 - 16
Settings and procedures prior to starting the operation ..... 4 - 3
Simultaneous broadcasting ..... 8 - 11
SMTP ..... 1 - 15
Specification ..... Chapter 3
Starting I/O No. ..... 4 - 17
Station No. ..... 4 - 18
Subheader
Fixed buffer communication (procedure exist) ..... 7 - 14
Communication using the random access buffer ..... 9 - 7
Sub-net address ..... 5 - 18
Sub-net mask pattern ..... 5 - 17
System configuration ..... Chapter 2

[T]
TCP ..... 1 - 14, 5 - 37
Test
Self refrain test ..... 4 - 22
Hardware test (H/W test) ..... 4 - 23
Loop back test ..... 5 - 29
PING command test ..... 5 - 34
Text (command)
Fixed buffer communication (procedure exist) ..... 7 - 15
Communication using the random access buffer ..... 9 - 9
Timer setting ..... 5 - 4

[U]
UDP ..... 1 - 15, 5 – 36
UINI instruction ..... 5 - 13, 10 - 23
Unpassive ..... 5 - 37, 5 - 52
Usage setting ..... 3 - 14, 5 - 36

[V]
Valid module during other station access ... 4 - 17

Index - 3

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.
However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.
Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not , compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. Product application

(1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
(2) The Mitsubishi programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable logic controller applications.
In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications.
However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

Microsoft, Windows, Windows NT are registered trademarks of Microsoft Corporation in the United States and other countries.
Netscape is a registered trademark of Netscape Communication Corporation in the United States and other countries.
Adobe and Acrobat are registered trademarks of Adobe Systems Incorporation.
Pentium and Celeron are trademarks of Intel Corporation in the United States and other countries.
Ethernet is a trademark of Xerox Co., Ltd. in the United States.
Other company names and product names used in this document are trademarks or registered trademarks of respective owners.

Q Corresponding Ethernet Interface Module

User's Manual (Basic)

MODEL	QJ71E71-U-KI-E
MODEL CODE	13JL88
SH(NA)-080009-I(0406)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212,JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.